RISERVATO PERSONAL

Spedizione in Abb. Postale Gruppo III/70
Questo fascicolo è annesso alla rivista Bit n. 50 Maggio 1984.

APPLE

FP.PLOT

Sovente chi deve affrontare col calcolatore elaborazioni complesse o raccolte di dati strumentali rimanda ad una fase successiva la visualizzazione in grafici dei dati ottenuti. Il programma FP.PLOT serve proprio per questa seconda fase di lavoro in quanto realizza grafici da dati memorizzati su disco.

di **Adriano Baracco**

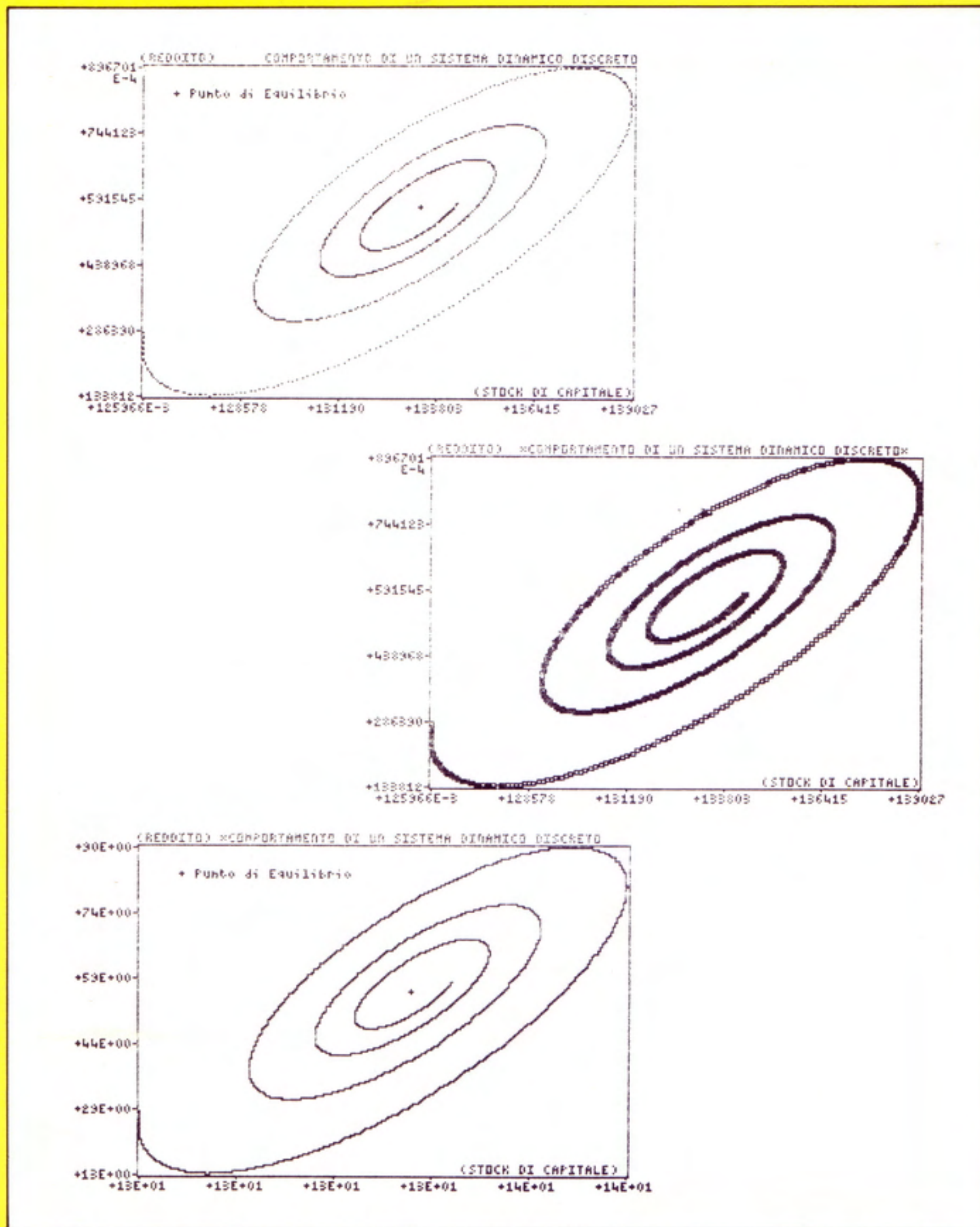
Le routine in linguaggio macchina

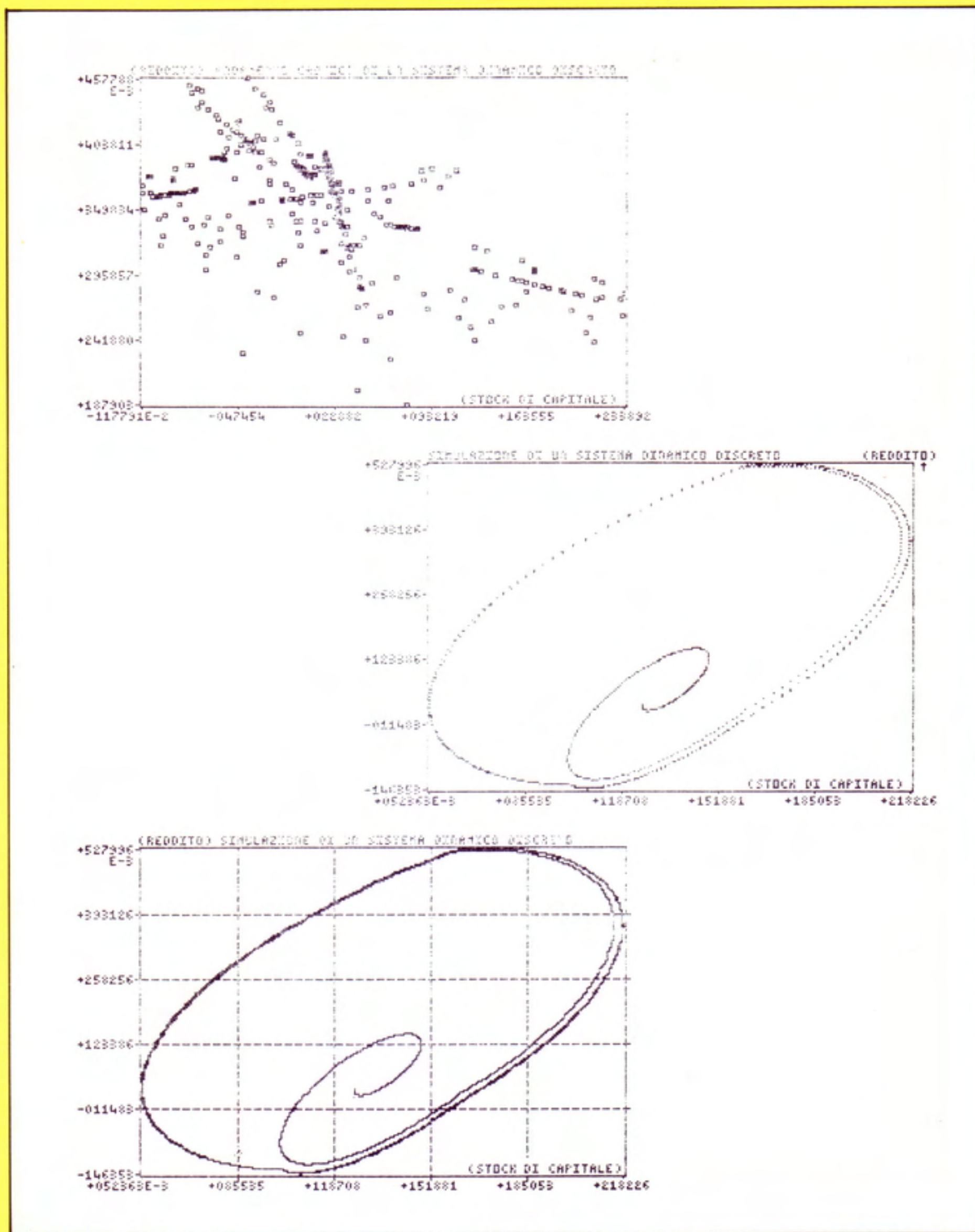
FP.PLOT comprende un programma BASIC che svolge quasi tutto il lavoro e due routine in linguaggio macchina: PRINT3★5 e FP.STORE.
La prima delle due routine, abbastanza classica, permette di scrivere sulla pagina grafica dell'Apple. Questa routine però si differenzia da quasi tutte le altre per l'uso di caratteri compresi in una matrice di tre punti per cinque. La scelta di queste dimensioni ridotte è dettata dall'esigenza di riservare il maggior spazio possibile al grafico vero e proprio pur mantenendo la possibilità di scrivervi a lato i valori di scala. La routine viene chiamata con una normale call cui segue dopo i due punti che separano le istruzioni in BASIC il comando di stampa che può essere o "PRINT" o "XPRINT"; la differenza consiste nel fatto che "XPRINT" scrive ogni punto del carattere col colore complementare a quello già presente sullo schermo. Quindi, per esempio, scrivendo due volte il medesimo carattere nella medesima posizione col comando "XPRINT" il carattere viene prima scritto e poi cancellato. Il testo da scrivere che deve seguire il comando "PRINT" o "XPRINT" può essere di tre tipi:
1) racchiuso tra " (virgolette) come per un normale print del BASIC;
2) rappresentato da una variabile alfanumerica ($);
3) racchiuso tra ' (apostrofi).
In questo ultimo caso i valori interni agli apostrofi devono essere numerici (numeri od espressioni) e se sono più di uno devono essere separati da virgole. Questi valori vengono considerati come codici ASC e la routine stampa i relativi caratteri.

Il testo non può essere più lungo di 70 caratteri (buffer di una riga), se questa condizione non viene rispettata il programma segnala errore nel caso di testi tra virgolette od apostrofi o scrive solo i caratteri contenibili nel buffer se il testo è rappresentato da una variabile alfanumerica (questo per evitare di dover ridurre la variabile stessa alle dimensioni giuste). Sempre nel limite dei settanta caratteri il testo può essere composto da più parti unite dal segno + (più). Infine in fondo al testo, separate da virgole, si possono indicare le coordinate del primo carattere (X,Y); se le coordinate non vengono indicate la routine scriverà il testo di seguito all'ultimo carattere scritto in precedenza. A questo proposito bisogna ricordarsi che il programma non testa durante la stampa se le coordinate vanno oltre i limiti della pagina grafica e questo può ovviamente creare dei problemi se non si calcola preventivamente la lunghezza del testo (ovviare a questo inconveniente porta però a rallentare in misura notevole la routine).
Riassumendo: se per esempio si vuole scrivere incominciando dal punto 10,10 la frase "prova stampa" con la routine PRINT3★5 si può procedere in uno dei seguenti modi:
1) CALL-28079: PRINT "PROVA STAMPA", 10, 10
2) A$ = "PROVA STAMPA": CALL-28079: PRINTA$, 10, 10
3) CALL-28079: PRINT' 80, 82, 79, 86, 65, 32, 83, 84, 65, 77, 80, 65', 10, 10
(oppure una combinazione di questi modi come ad esempio: CALL-28079: PRINT "PROVA" + '83, 84, 65, 77, 80, 65', 10, 10)
Ovviamente gli stessi comandi valgono per "XPRINT". È ammessa, inoltre, la notazione sintetica "?" per "PRINT" e "X?" per "XPRINT".

Il set di caratteri occupa complessivamente 256 byte per un totale di 128 caratteri (2 byte per carattere) e comprende i 96 caratteri ASC (codici da 32 a 127) più 32 caratteri speciali (codici da 0 a 31), di cui i primi dieci sono usati dal programma per la realizzazione dei grafici per simboli. I caratteri sono definiti partendo dal primo punto in alto (bit 0 del primo byte) per colonne verticali (ultimo punto = bit 6 del secondo byte).
Tutte queste spiegazioni sono necessarie solo per chi eventualmente voglia usare la routine al di fuori del programma in quanto all'interno di esso la routine è gestita automaticamente.
La seconda routine serve invece per tradurre un numero in virgola mobile in una stringa di 7 caratteri e, ovviamente, per la traduzione inversa. Ciò permette di memorizzare su disco i dati FP in file di testo ad accesso casuale di 8 byte (7 + il return) con un considerevole risparmio di memoria se si pensa che un dato FP può occupare anche 16 byte (+X.XXXXXXXXE+XX). I numeri FP occupano in applesoft 5 byte da 8 bit ciascuno, il DOS però non memorizza nei file di testo l'ottavo bit e quindi i byte devono essere ridotti a 7 bit. Anche così rimane tuttavia il problema di valori come il CHR$(4) che viene considerato un comando DOS o del CHR$(0) che non esiste. Ponendo ad 1 il settimo bit anche questo secondo problema è risolto ma i byte necessari per codificare un numero FP passano da 5 (di tipo XXXXXXXX) a 7 di tipo (01XXXXXX).

Per operare la trasformazione da FP a stringa è sufficiente chiamare la routine con la solita call seguita (dopo i due punti) dal nome della variabile alfanumerica in cui va memorizzata la stringa, dal segno = (uguale) ed infine da una variabile o da una espressione numerica (es. CALL-

Figura 1 - Esempi di grafici: per punti, per simboli, per linee.

27367: AA$ = 5★SIN(3)★1). Analogamente per la trasformazione inversa alla CALL deve seguire il nome della variabile FP, il segno = ed infine il nome della variabile alfanumerica. Nel caso di conversione da FP a stringa, poiché la routine sostituisce i valori ORIGINALI della stringa con quelli calcolati, è necessario che la variabile alfanumerica contenga già sette caratteri (occorre cioè "dimensionare" la variabile con un comando del tipo AA$ = "1234567", in cui ovviamente la stringa di 7 caratteri può contenere qualsiasi messaggio). Qualora la stringa sia di lunghezza diversa da quella richiesta la routine interromperà il programma con il messaggio di errore "STRING TOO LONG". I messaggi di errore delle due routine in linguaggio macchina sono facilmente distinguibili da quelle BASIC perché vengono visualizzate in modo diverso.
Per quanto concerne il funzionamento delle due routine questo è spiegato nel listato commentato ottenuto con l'Assembly LISA 2.0.
Chi non possiede il LISA può comunque copiare i valori esadecimali che sono riportati in tabella con i valori di somma che servono per controllare se l'impostazione è stata corretta (ad esempio: dopo aver impostato i valori di FP.STORE da $9519 a $95FF si calcola con il solito ciclo BASIC "S=0: FOR I = −27367 TO −27137: S = S + PEEK(I): next" il

Seguito figura 1.

valore della somma (S) che deve risultare uguale a 231; in caso contrario si sono verificati errori di copiatura).

Il programma carica le due routine come un unico file e quindi una volta inseriti i valori delle due routine e della tabella dei caratteri da $9251 a $95ff si deve memorizzare il tutto su disco con il comando DOS "BSAVE FP.PLOT.BIN, A$9251, L$3AF".È comunque conveniente registrare i due programmi anche separatamente con i comandi "BSAVE FP.STORE,A$9519,L$EF" e "BSAVE PRINT3★5, A$9521, L$2C8". La routine FP.STORE dovrà infatti essere usata per la creazione dei file dati. La memoria complessivamente usata dalle due routine, dalla tabella dei caratteri e dal buffer della routine di PRINT va da $9206 a $95FF. L'HIMEM viene settato quindi a $9205 (−28155). La memoria è così distribuita:

$9206-$9250: buffer + variabili di PRINT 3★5
$9250-$9418: PRINT 3★5%
$9419-$9518: TABELLA DEI CARATTERI
$9519-$95FF: FP.STORE

Nella mappa in figura 4 sono riportati anche i valori decimali.

Per i programmi che usano solo la routine FP.STORE (quelli di creazione file dati) l'himem deve

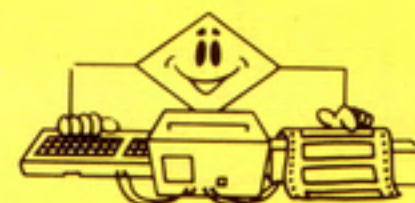
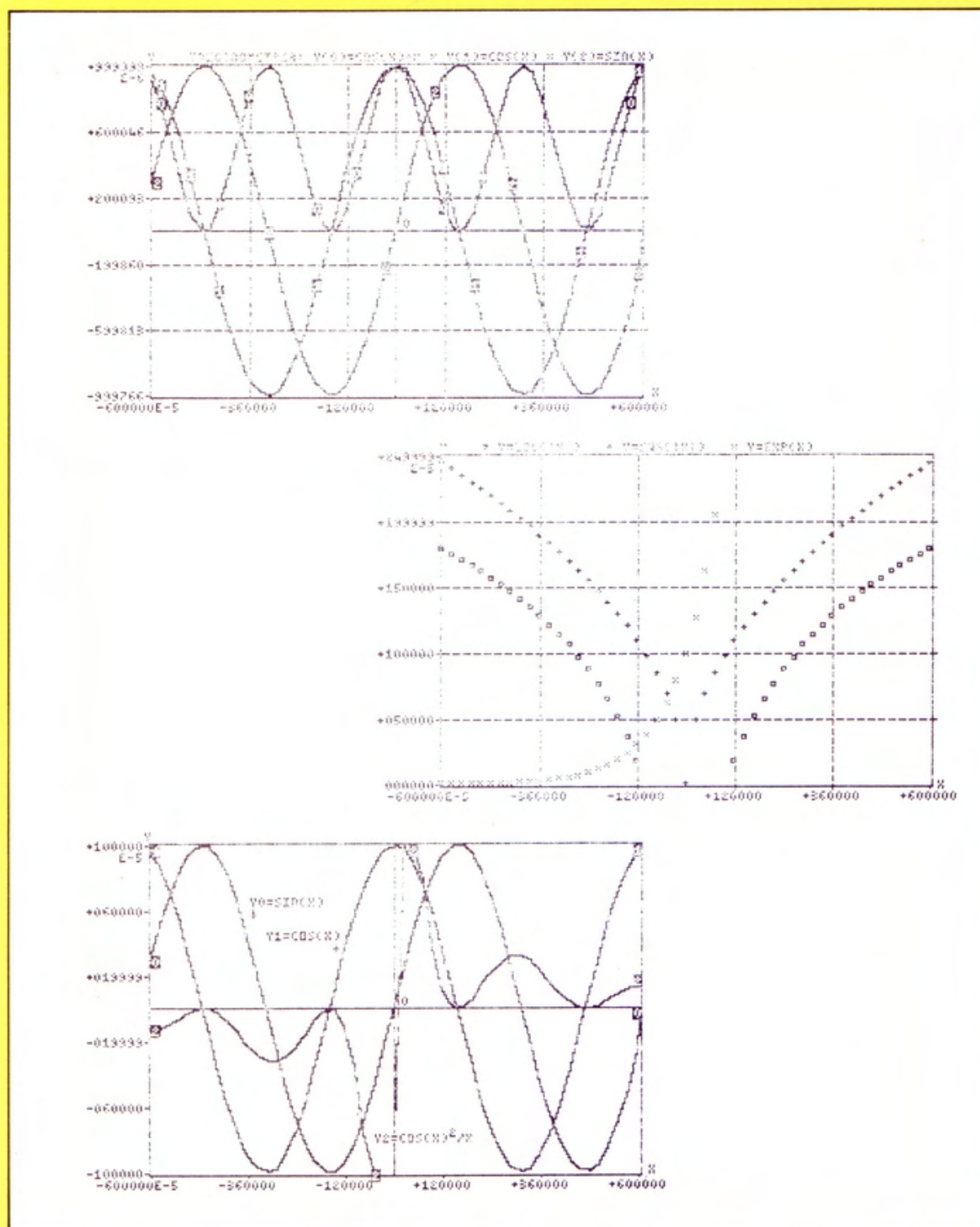



Figura 2 - Esempi di grafici: con griglia e assi, solo con griglia, solo con assi.

essere fissato al valore \$9518 (−27368).

L'uso del programma

Alcune convenzioni per le fasi di input vengono rispettate per tutto il programma:
1) quando il cursore si trova tra parentesi quadre la risposta viene accettata con un GET e quindi non occorre premere il RETURN;
2) al termine di ogni ciclo di risposte il programma chiede conferma dei dati impostati;
3) quando ad una richiesta di un dato si risponde con il RETURN il programma interpreta la risposta come una conferma al valore già posseduto da quel dato. Il RETURN viene quindi accettato solo se il programma ha già avuto in precedenza una risposta a quella richiesta (e la risposta viene confermata) oppure se conosce già - perché lo ha calcolato - il valore del dato (valore che l'operatore non intende cambiare).

Le opzioni

La prima serie di opzioni viene presentata dal

Figura 3 - Tabella dei comandi disponibili.

TABELLA DEI COMANDI DISPONIBILI (OPZIONE 5)

MODO	CURSORE	ACCESSO	FUNZIONI
(M)UOVI	M	ESC M	I = MUOVE IN ALTO J = MUOVE A SINISTRA K = MUOVE A DESTRA M = MUOVE IN BASSO
(S)CRIVI (X)SCRIVI [illegible] [illegible]	S X [illegible] X	ESC S ESC X [illegible] [illegible]	SCRIVE IN MAIUSCOLO SCRIVE-CANCELLA (MAIUSCOLO) SCRIVE IN MINUSCOLO SCRIVE-CANCELLA (MINUSCOLO)
COMANDI	[illegible]	ESC	X = ENTRA IN MODO XSCRIVI S = ENTRA IN MODO SCRIVI [illegible] = MINUSCOLO [illegible] = MAIUSCOLO [illegible] = MODO CARATTERI SPECIALI M = MODO MUOVI [illegible] = MODO RICERCA PUNTI ? = SCRIVE IL PUNTO TROVATO [illegible] = MODO INPUT VELOCITA'
RICERCA PUNTI [illegible]	[illegible]	ESC	PERMETTE DI POSIZIONARE IL CURSORE SU UN PUNTO (COME M)
CARATTERI SPECIALI[illegible]		ESC C	[illegible] = [illegible] IL CARATTERE RETURN = SCRIVE IL CARATTERE
(V)ELOCITA'	?	ESC V	INPUT VELOCITA' DEL CURSORE

COMANDI DISPONIBILI IN MODO: SCRIVI, XSCRIVI, scrivi, xscrivi

1) RETURN (CTRL-M) : SCENDE DI UNA RIGA
2) [illegible] : SALE DI UNA RIGA
3) <- (CTRL-H) : CANCELLA
4) -> (CTRL-U) : RISCRIVE

MAX 255 CARATTERI DI BUFFER

Figura 4 - Mappa di memoria del programma.

programma dopo l'input del nome delle curve (i file dati). A questo proposito, se durante l'input si verifica un errore, battendo al posto del nome della curva il CTRL-E e il RETURN, l'operatore potrà riscrivere il nome dell'ultima curva impostata. Ad esempio, se al quarto nome ci si accorge di aver sbagliato l'impostazione del secondo file, si procede così:

1) CTRL-E RET
2) CTRL-E RET
3) (correzione) RET
4) RET
5) ....

In ogni caso, la fase d'input si conclude con la richiesta di conferma di tutti i dati impostati.

Le opzioni cui ho poco sopra accennato sono:

(G) GRIGLIA: permette di sovrapporre al grafico un reticolo di riferimento ai valori di scala

(S) ASSI IN SCALA: permette di scegliere per i valori di scala tra il formato +XXE+XX (in cui ogni valore è in notazione esponenziale, a scapito del numero di cifre) ed il formato + XXXXXX (in cui tutti i valori sugli assi sono moltiplicati per uno stesso esponente indicato per la X dopo il primo

valore a sinistra e per la Y sotto il primo valore in alto). Questa seconda notazione, che utilizza sei cifre significative, è usata quando questa opzione (S) è attivata.
(R) REGISTRA MAX/MIN: quando questa opzione è ON il programma - trovati i massimi ed i minimi di una funzione - li registra al fondo del file.
(A) DISEGNA GLI ASSI: segna gli assi cartesiani (origine in 0,0)
Seguono tre opzioni alternative che permettono anche di uscire da questa fase di input (vanno quindi selezionate dopo aver scelto le precedenti opzioni). Per distinguere queste opzioni dalle precedenti (che sono ad interruttore) l'alternativa non è posta tra lettere ma tra numeri:
(1) GRAFICO PER PUNTI
(2) GRAFICO PER LINEE
(3) GRAFICO PER SIMBOLI
Nei primi due casi, al termine della realizzazione del grafico, l'operatore può numerare le curve. I simboli della terza opzione sono i primi dieci caratteri del set di PRINT 3★5 (dieci è il numero massimo di curve disegnabili contemporaneamente).

Scelte le opzioni l'operatore può impostare una intestazione per il grafico (massimo 55 caratteri) e una intestazione per gli assi (massimo tre caratteri per ogni asse). Per saltare questa fase è sufficiente rispondere con un RETURN ai tre input. Se invece si imposta un'intestazione di lunghezza maggiore di quella consentita il programma farà lampeggiare una freccia (ˆ) sotto l'ultimo carattere accettabile: premendo RETURN il programma userà come intestazione il messaggio dall'inizio del messaggio stesso fino alla freccia lampeggiante.
A questa fase segue la ricerca dei minimi e dei massimi della curva. Se questa è già stata fatta in precedenza con l'opzione R attiva il programma si limiterà a leggere i massimi ed i minimi su disco. Trovati tutti gli estremi delle curve il programma - dopo averli scritti in tabella - chiede all'operatore se intende cambiarli. In caso affermativo vengono richiesti i nuovi estremi. Come al solito, qualora l'operatore non voglia cambiare un estremo, deve rispondere con il RETURN. È possibile, inoltre, rispondere al primo input (valore minimo di x) con una C seguita dal numero di una delle curve (esempio C4 per la curva n.ro 4), il programma allora userà come estremi quelli della curva indicata. Stabiliti gli estremi, il programma procede alla realizzazione del grafico.

Terminato il grafico, se si sono scelte le opzioni "grafico per punti" (1) o "per linee" (2), l'operatore può numerare le curve rispondendo affermativamente alla relativa richiesta.
Viene in questo caso presentata la pagina grafica con un cursore lampeggiante. L'operatore può muovere il cursore lungo la prima curva premendo le due frecce della tastiera; raggiunto il punto in cui si intende numerare la curva, il RETURN rende operativa questa opzione. Una volta numerata la curva, premendo ESC si passa alla curva successiva. Il programma prima di passare alle curve successive chiede conferma facendo lampeggiare la scritta "OK?" ed attendendo la risposta (S o N). In caso di risposta negativa il cursore torna a lampeggiare sull'ultima curva considerata. Qualora il punto in cui il cursore si trova sia fuori scala il programma fa lampeggiare una F. Esaurite le curve il programma farà lampeggiare la scritta ESC per segnalare all'operatore la fine del lavoro; ESC fa uscire in modo testo (questa operazione è standard tutte le volte che, terminato un lavoro, il programma deve tornare in modo testo).
A questo punto si presenta un nuovo menu di operazioni:
1) REGISTRA IL GRAFICO: chiede il nome del grafico e lo registra su disco;
2) ALTRE CURVE: riparte dall'input del nome delle curve da disegnare;
3) CAMBIA LA SCALA: riscrive la tabella degli estremi delle curve, ponendo in inverso i massimi ed i minimi globali; in fondo alla tabella scrive gli ultimi estremi usati per il disegno e passa all'input dei nuovi;
4) CAMBIA TIPO DI GRAFICO: torna all'input della prima serie di opzioni;
5) SCRITTE: la descrizione nel paragrafo successivo;
6) FINE.

Possibilità di scrittura

Con l'opzione SCRITTE (5) si accede ad una serie di funzioni di scrittura diretta sulla pagina grafica. Il programma si dispone in modo testo con come cursore una M lampeggiante. Ciò significa che ci si trova in modo "MUOVI" (vedi tabella). Premendo le lettere I, J, K, M ci si può spostare col cursore rispettivamente in alto, a sinistra, a destra ed in basso. La velocità di movimento è modificabile premendo i tasti ESC e V; il cursore diventa "?" ed il programma si predispone all'input della velocità (massimo due cifre), dopo di che ritorna al modo M. Dopo aver posizionato il cursore dove si desidera, si può passare in modo SCRIVI o XSCRIVI premendo ESC seguito rispettivamente da S o X; il cursore diventa appunto "S" o "X". L'operatore è così in grado di scrivere direttamente sulla pagina grafica (da notare che usando il modo X si può una volta scritto un messaggio cancellarlo riottenendo lo sfondo originale).
1) RETURN: il cursore si abbassa di una riga;
2) CTRL-I: il cursore si alza di una riga;
3) FRECCIA A SINISTRA: cancella (e memorizza in un buffer di 255 caratteri);
4) FRECCIA A DESTRA: riscrive;
5) ESC-L: passa in minuscolo (cursore "x" o "s");
6) ESC-U: torna in maiuscolo;
7) ESC-@: passa alla visualizzazione di caratteri speciali; che sono selezionabili con le due frecce. Il carattere viene scelto col RETURN, mentre ESC permette di uscire senza la stampa del carattere.

Un terzo modo è accessibile premendo ESC. (punto). Il cursore diventa una crocetta che si può spostare, come nel modo M, con i tasti I, J, M, K. Posizionato il cursore sul punto desiderato, l'operatore può ottenere le coordinate del punto premendo i tasti ESC? (punto interrogativo). Premendo RETURN il programma visualizza nuovamente la pagina grafica con il cursore (crocetta) lampeggiante. Per passare da un modo all'altro in ogni momento è sufficiente premere il tasto ESC seguito dal tasto che simboleggia il modo che si vuol attivare (S per scrivi, M per muovi, ecc.).
ESQ Q, infine, permette di uscire dall'opzione

APPLE

SCRIVI e di tornare al menu principale.

I file dati

I file trattati dal programma sono file casuali (lunghezza = 8) che originariamente devono contenere:

- nel record 0: il numero di punti (n);
- nei record da 1 a 2★n+1: le n coppie di coordinate (alternativamente la x e la y).

Dopo l'uso con l'opzione R attiva il file conterrà:

- nel record 0: il numero di punti (n);
- nei record da 1 a 2★n+1: invariato;
- nel record 2★n+2: il valore di x minimo;
- nel record 2★n+3: il valore di x massimo;
- nel record 2★n+4: il valore di y minimo;
- nel record 2★n+5: il valore di y massimo.

I dati contenuti nel file dovranno essere del tipo restituito dalla routine FP.STORE. Avendo però dati memorizzati normalmente (non più di tremila punti), si può usare il piccolo programma TRASFORMAZIONE DATI per creare file manipolabili da FP.PLOT.

REMarks al programma

100-130: subroutine che scrive gli estremi di una curva.
140-160: subroutine che converte la variabile numerica AA nel formato "+XXE+XX" (nella variabile alfanumerica AA$).
170-180: subroutine che converte AA nella stringa AA$ (formato "+XXXXXX").
190-240: subroutine che scrive l'intestazione della tabella massimi-minimi.
250-330: subroutine che scrive gli estremi della curva in fondo al file e pone in negativo il numero di punti per segnalare al programma che questo file ha memorizzati i massimi ed i minimi.
340-350: subroutine che memorizza i caratteri scritti sulla pagina grafica.
1000-1140: inizializza le costanti e carica i programmi in l.m. In particolare la riga 1040 trova lo slot usato dal DOS e calcola i valori di accensione (SI%) e spegnimento (NO%) del giradischi; questi valori vengono usati in seguito per mantenere il giradischi in rotazione velocizzando la lettura dei dati. A questo proposito: se si interrompe il programma durante la lettura dati il giradischi continua a girare finché non si preme il RESET o non si scrive un comando POKE o PEEKE che faccia riferimento alla locazione NO%. Se si vuole evitare questo inconveniente è sufficiente aggiungere al programma i seguenti due passi:
0 ONERR GO TO 5000
5000 POKE NO%,0: TEXT: HOME: PRINT "errore nr.o-"PEEK(222)" alla linea "PEEK(218) + PEEK(219)★256
1150-1340: input del numero del drive contenente il disco dati ed input del nome delle curve da disegnare.
1350-1770: input delle opzioni per il grafico e delle intestazioni.
1780-2390: calcolo degli estremi delle curve ed eventuale input dei nuovi estremi.
2400-2630: disegno dello sfondo (assi, griglia, valori di scala, intestazione, ecc.).
2640-2870: traccia le curve.
2880-3260: numerazione delle curve e OK? lampeggiante.
3270-3300: ESC lampeggiante.
3310-3450: menu finale.
3460-3490: registrazione del grafico su disco.
3500-3630: creazione tabella degli estremi per il cambio di scala.
3640-3690: cursore lampeggiante ed input per l'opzione 5.
3700-3730: salti ai vari "modi".
3740-3780: I, J, K, M delle routine "muovi" e "ricerca punti".
3790-3830: controllo delle coordinate di stampa.
3840-3910: input velocità cursore.
3920-4130: routine di scrittura.
4140-4220: routine di ESC.
4230-4380: caratteri speciali.
4390-4480: stampa le coordinate del punto (modo ricerca punti).

Listato 1 - Il listato di FP.PLOT.

```
10  GOTO 1000
100 AA = W1(Z): GOSUB 140: VTAB 8 + I: HTAB 9: PRINT AA$
110 AA = M1(Z): GOSUB 140: VTAB 8 + I: HTAB 17: PRINT AA$
120 AA = W2(Z): GOSUB 140: VTAB 8 + I: HTAB 25: PRINT AA$
130 AA = M2(Z): GOSUB 140: VTAB 8 + I: HTAB 33: PRINT AA$: RETURN
140  IF  NOT (AA) THEN AA$ = "+00E+00": RETURN
150 AA$ = S$( SGN (AA) + 1):E =  LOG ( ABS (AA)) / LG:E =  INT (E * 1E7 +
    1) * 1E - 7:E% = E:E = E - E%:AA =  INT (10 ^ E * 10 + .5): IF AA = 1
    00 THEN AA = 10:E% = E% + 1
160 E% = E% - 1:AA$ = AA$ +  STR$ (AA) + "E" + S$(1 +  SGN (E%)):E% =  ABS
    (E%):AA$ = AA$ + SP$(E% > 9) +  STR$ (E%): RETURN
170  IF AA = 0 THEN AA$ = " 000000": RETURN
180 AA$ = S$( SGN (AA) + 1):E =  LOG ( ABS (AA)) / LG:E = E + DE%:AA =  INT
    (10 ^ E):AA$ = AA$ +  RIGHT$ ("000000" +  STR$ (AA),6): RETURN
190  VTAB 3: PRINT "****************************************"
200  VTAB 4: PRINT "*" SPC( 6)"*" SPC( 7)"*" SPC( 7)"*" SPC( 7)"*" SPC( 7
    )"*"
210  VTAB 5: PRINT "*PUNTI * X MIN * X MAX * Y MIN * Y MAX *"
220  VTAB 6: PRINT "*" SPC( 6)"*" SPC( 7)"*" SPC( 7)"*" SPC( 7)"*" SPC( 7
    )"*"
230  VTAB 7: PRINT "****************************************"
240  RETURN
250  PRINT D$"WRITE"C$(I)",R"NP * 2 + 1
260  CALL CV%:TR$ = W1(0): PRINT TR$
270  CALL CV%:TR$ = M1(0): PRINT TR$
```

Seguito listato 1.

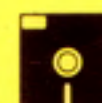
```
280  CALL CV%:TR$ = W2(0): PRINT TR$
290  CALL CV%:TR$ = M2(0): PRINT TR$
300  PRINT D$"WRITE"C$(I)",R0"
310  CALL CV%:TR$ =  - NP: PRINT TR$
320  PRINT D$"CLOSE"C$(I)
330  RETURN
340  IF  LEN (B1$) < 255 THEN B1$ = B1$ +  CHR$ (K%): RETURN
350 B1$ =  RIGHT$ (B1$,254) +  CHR$ (K%): RETURN
1000  HIMEM:  - 28155
1010  LOMEM: 24576
1020  DIM C$(9),XM(9),XW(9),YM(9),YW(9),S$(2),SP$(1),ST$(1),M1(1),M2(1),W
    1(1),W2(1)
1030 D$ =  CHR$ (4):I$ =  CHR$ (91) + " ]":TR$ = "1234567"
1040 NO% =  - 16248 +  PEEK (1528):SI% = NO% + 1
1050  PRINT D$"BLOAD FP.PLOT.BIN"
1060 S$(0) = "-":S$(1) = "+":S$(2) = "+":SP$(0) = "0":ST$(0) = "OFF":ST$(
    1) = "ON "
1070 PR% =  - 28079
1080 LG =  LOG (10)
1090 CV% =  - 27367
1100 G% = 1:SC% = 1:WR% = 0:AS% = 1
1110 KR% =  - 16384:KC% =  - 16368
1120 SPEAKER% =  - 16336
1130 SG% =  - 16304:FS% =  - 16302:HG% =  - 16297:P2% =  - 16299:E6% = 23
    0:BK% =  - 3082
1140 TI% = 20
1150 FF$ = "": TEXT : HOME
1160  POKE KC%,0: VTAB 5: PRINT "I DATI SONO SUL DRIVE "I$
1170  VTAB 5: HTAB 24: GET Z$: IF Z$ =  CHR$ (13) AND D$ <  > "" THEN Z$ =
    RIGHT$ (DR$,1)
1180  PRINT Z$: IF Z$ <  > "1" AND Z$ <  > "2" THEN  PRINT  CHR$ (7): GOTO
    1170
1190  POKE KC%,0:DR$ = ",D" + Z$
1200  VTAB 10: INPUT "NUMERO DI CURVE (MAX 10) ? ";Z$: IF Z$ = "" AND NC%
    THEN  VTAB 10: HTAB 28: PRINT NC%: GOTO 1230
1210  IF  VAL (Z$) <  = 0 OR  VAL (Z$) > 10 THEN  PRINT  CHR$ (7): GOTO 1
    200
1220 NC% =  VAL (Z$)
1230  HOME : VTAB 2: PRINT "* NOME DELLE CURVE : "
1240  FOR I = 0 TO NC% - 1
1250  VTAB 5 + I: PRINT "CURVA  "I" ="
1260  VTAB 5 + I: HTAB 12: INPUT "";Z$: IF Z$ = "" AND C$(I) = "" THEN  PRINT
    CHR$ (7): GOTO 1260
1270  IF Z$ = "" THEN 1300
1280  IF Z$ =  CHR$ (5) THEN I = I - 1:I = I * (I > 0): GOTO 1250
1290 C$(I) = Z$
1300  VTAB 5 + I: HTAB 12: PRINT C$(I)
1310  NEXT I
1320  POKE KC%,0: VTAB 20: PRINT  CHR$ (7)"TUTTO CORRETTO ? "I$
1330  VTAB 20: HTAB 19: GET Z$: PRINT Z$: IF Z$ <  > "S" AND Z$ <  > "N" THEN
    PRINT  CHR$ (7): GOTO 1330
1340  IF Z$ = "N" THEN 1150
1350  HOME
1360  VTAB 5: HTAB 4: PRINT "(G) GRIGLIA............."
1370  VTAB 7: HTAB 4: PRINT "(S) ASSI IN SCALA......."
1380  VTAB 9: HTAB 4: PRINT "(R) REGISTRA MAX/MIN...."
1390  VTAB 11: HTAB 4: PRINT "(A) DISEGNA GLI ASSI"
1400  VTAB 13: HTAB 4: PRINT "(1) GRAFICO PER PUNTI"
1410  VTAB 15: HTAB 4: PRINT "(2) GRAFICO PER LINEE"
1420  VTAB 17: HTAB 4: PRINT "(3) GRAFICO PER SIMBOLI"
1430  POKE KC%,0: VTAB 20: HTAB 15: PRINT I$
1440  VTAB 5: HTAB 28: INVERSE : PRINT ST$(G%): VTAB 7: HTAB 28: PRINT ST
    $(SC%): VTAB 9: HTAB 28: PRINT ST$(WR%): VTAB 11: HTAB 28: PRINT ST$(
    AS%): NORMAL
1450  VTAB 20: HTAB 16: GET Z$: PRINT Z$: IF Z$ =  CHR$ (13) AND  NOT FL%
    THEN  CALL  - 198: GOTO 1450
1460  IF Z$ =  CHR$ (13) THEN 1530
1470  IF Z$ = "G" THEN G% = G% = 0: GOTO 1440
1480  IF Z$ = "S" THEN SC% = SC% = 0: GOTO 1440
1490  IF Z$ = "R" THEN WR% = WR% = 0: GOTO 1440
1500  IF Z$ = "A" THEN AS% = AS% = 0: GOTO 1440
1510  IF Z$ < "1" OR Z$ > "3" THEN  CALL  - 198: GOTO 1450
1520 FL% =  VAL (Z$)
1530  VTAB 20: HTAB 16: PRINT FL%
1540  HOME
```

APPLE

Seguito listato 1.

```
1550  VTAB 2: PRINT "*** INTESTAZIONE  DEL  GRAFICO   ***"
1560  VTAB 6: PRINT "*TITOLO (MAX 55 CARATTERI)"
1570  POKE 34,6: HOME : INVERSE
1580  VTAB 9: HTAB 15: PRINT "^": NORMAL
1590  VTAB 7: INPUT "";Z$
1600  IF Z$ <  > "" THEN TI$ = Z$
1610  IF  LEN (TI$) > 55 THEN TI$ =  LEFT$ (TI$,55): CALL  - 198: FLASH :
    GOTO 1580
1620  HOME : PRINT TI$: VTAB 10: PRINT "* ASSE X (MAX 3 CARATTERI)"
1630  POKE 34,10: HOME : INVERSE
1640  VTAB 12: HTAB 3: PRINT "^": NORMAL
1650  VTAB 11: INPUT "";Z$: IF Z$ <  > "" THEN XX$ = Z$
1660  IF  LEN (XX$) > 3 THEN XX$ =  LEFT$ (XX$,3): CALL  - 198: FLASH : GOTO
   1640
1670  HOME : PRINT XX$: VTAB 14: PRINT "* ASSE Y (MAX 3 CARATTERI)"
1680  POKE 34,14: HOME : INVERSE
1690  VTAB 16: HTAB 3: PRINT "^": NORMAL
1700  VTAB 15: INPUT "";Z$: IF Z$ <  > "" THEN YY$ = Z$
1710  IF  LEN (YY$) > 3 THEN YY$ =  LEFT$ (YY$,3): CALL  - 198: FLASH : GOTO
   1690
1720  HOME : PRINT YY$
1730  POKE KC%,0: VTAB 20: PRINT  CHR$ (7)"CONFERMI ? ";I$
1740  TEXT
1750  VTAB 20: HTAB 13: GET Z$: PRINT Z$: IF Z$ <  > "S" AND Z$ <  > "N" THEN
    CALL  - 198: GOTO 1750
1760  IF Z$ = "N" THEN 1350
1770  IF FF$ = "!" THEN 3500
1780 FF$ = "!": HOME : FLASH
1790  VTAB 1: PRINT "*  RICERCA DEI MASSIMI E DEI MINIMI   *"
1800  NORMAL : GOSUB 190
1810  FOR I = 0 TO NC% - 1
1820  PRINT D$"VERIFY"C$(I)DR$
1830  PRINT D$"OPEN "C$(I)",L8"
1840  PRINT D$"READ "C$(I)
1850  INPUT TR$: CALL CV%:NP = TR$: IF NP >  = 0 THEN 1900
1860 NP =  - NP: PRINT D$"READ"C$(I)",R"NP * 2 + 1
1870  INPUT TR$: CALL CV%:W1(0) = TR$: INPUT TR$: CALL CV%:M1(0) = TR$: INPUT
   TR$: CALL CV%:W2(0) = TR$: INPUT TR$: CALL CV%:M2(0) = TR$
1880  VTAB I + 8: HTAB 3: PRINT NP
1890  GOTO 2010
1900  INPUT TR$: CALL CV%:X = TR$: INPUT TR$: CALL CV%:Y = TR$
1910 M1(0) = X:M2(0) = Y:W1(0) = X:W2(0) = Y
1920  VTAB I + 8: HTAB 3: PRINT 1
1930  FOR J = 2 TO NP: VTAB I + 8: HTAB 3: PRINT J
1940  INPUT TR$: CALL CV%:X = TR$: INPUT TR$: CALL CV%:Y = TR$
1950  IF M1(0) < X THEN M1(0) = X
1960  IF W1(0) > X THEN W1(0) = X
1970  IF M2(0) < Y THEN M2(0) = Y
1980  IF W2(0) > Y THEN W2(0) = Y
1990 A% =  PEEK (SI%)
2000  NEXT J: IF WR% THEN  GOSUB 250
2010  IF  NOT I THEN W1(1) = W1(0):M1(1) = M1(0):M2(1) = M2(0):W2(1) = W2
   (0)
2020 Z = 0: GOSUB 100
2030 XW(I) = W1(0):YW(I) = W2(0):XM(I) = M1(0):YM(I) = M2(0)
2040  IF M1(1) < M1(0) THEN M1(1) = M1(0)
2050  IF M2(1) < M2(0) THEN M2(1) = M2(0)
2060  IF W1(1) > W1(0) THEN W1(1) = W1(0)
2070  IF W2(1) > W2(0) THEN W2(1) = W2(0)
2080  PRINT D$"CLOSE"C$(I)
2090  NEXT I:A% =  PEEK (NO%)
2100  VTAB 1: PRINT "*  RICERCA DEI MASSIMI E DEI MINIMI   *"
2110 W1(0) = W1(1):W2(0) = W2(1):M2(0) = M2(1):M1(0) = M1(1)
2120  VTAB 20: PRINT "GLOBALE"
2130 Z = 1:I = 12: GOSUB 100
2140  POKE KC%,0: POKE 34,21: HOME : VTAB 22: PRINT  CHR$ (7)"VUOI CAMBIA
   RE QUESTI LIMITI ? "I$
2150  VTAB 22: HTAB 32: GET Z$: PRINT Z$;: IF Z$ <  > "S" AND Z$ <  > "N"
    THEN  PRINT  CHR$ (7): GOTO 2150
2160  IF Z$ = "N" THEN 2400
2170  HOME
2180  VTAB 22: INPUT "* X MIN = ";Z$
2190  IF Z$ = "" THEN 2260
2200  IF  ASC (Z$) <  > 67 OR  LEN (Z$) <  > 2 THEN 2250
2210 Z$ =  RIGHT$ (Z$,1)
```

Seguito listato 1.

```
2220 A% =  VAL (Z$): IF A% < Ø OR A% > NC% - 1 THEN 218Ø
223Ø W1(1) = XW(A%):W2(1) = YW(A%):M1(1) = XM(A%):M2(1) = YM(A%)
224Ø Z = 1: GOSUB 1ØØ: GOTO 214Ø
225Ø W1(1) =  VAL (Z$)
226Ø AA = W1(1): GOSUB 14Ø: VTAB 2Ø: HTAB 9: PRINT AA$
227Ø  HOME
228Ø  VTAB 22: INPUT "* X MAX = ";Z$: IF Z$ = "" THEN 23ØØ
229Ø M1(1) =  VAL (Z$)
23ØØ AA = M1(1): GOSUB 14Ø: VTAB 2Ø: HTAB 17: PRINT AA$
231Ø  HOME
232Ø  VTAB 22: INPUT "* Y MIN = ";Z$: IF Z$ = "" THEN 234Ø
233Ø W2(1) =  VAL (Z$)
234Ø AA = W2(1): GOSUB 14Ø: VTAB 2Ø: HTAB 25: PRINT AA$
235Ø  HOME
236Ø  VTAB 22: INPUT "* Y MAX = ";Z$: IF Z$ = "" THEN 238Ø
237Ø M2(1) =  VAL (Z$)
238Ø AA = M2(1): GOSUB 14Ø: VTAB 2Ø: HTAB 33: PRINT AA$
239Ø  GOTO 214Ø
24ØØ  TEXT : HOME : FLASH : VTAB 2: HTAB 5: PRINT "*   CREAZIONE  DELLO
    SFONDO   *"
241Ø  NORMAL
242Ø E1% = 6:E2% = 6: IF SC% AND M1(1) THEN J =  LOG ( ABS (M1(1))) *  SGN
    (M1(1)) / LG:E1% = E1% - J - (J = Ø)
243Ø  IF SC% AND M2(1) THEN J =  LOG ( ABS (M2(1))) *  SGN (M2(1)) / LG:E
   2% = E2% - J - (J = Ø)
244Ø FX = 235 / (M1(1) - W1(1)):FY = 175 / (M2(1) - W2(1))
245Ø  POKE E6%,64: POKE 28,127: CALL BK%
246Ø  HCOLOR= Ø: HPLOT 3Ø,6 TO 3Ø,183 TO 267,183 TO 267,6 TO 3Ø,6: FOR I =
    7 TO 182 STEP 35: HPLOT 28,I TO 29,I: NEXT I
247Ø  IF  NOT AS% THEN 252Ø
248Ø AX% = 31 - W1(1) * FX: IF AX% < 31 OR AX% > 266 THEN AX% = Ø
249Ø  IF AX% THEN  HPLOT AX%,7 TO AX%,182
25ØØ AY% = 182 + W2(1) * FY: IF AY% < 7 OR AY% > 182 THEN AY% = Ø
251Ø  IF AY% THEN  HPLOT 31,AY% TO 266,AY%
252Ø  FOR I = 31 TO 266 STEP 47: HPLOT I,183 TO I,184: NEXT
253Ø  CALL PR%: PRINT TI$,48,1
254Ø  CALL PR%: PRINT YY$,28,1: CALL PR%: PRINT XX$,269,18Ø
255Ø DE% = E2%:YS = 18Ø:ST = (M2(1) - W2(1)) / 5: FOR I = W2(1) TO M2(1) STEP
    ST:AA = I: ON SC% + 1 GOSUB 14Ø,17Ø: CALL PR%: PRINT AA$,Ø,YS:YS = YS
     - 35: NEXT I
256Ø AA = M2(1): ON SC% + 1 GOSUB 14Ø,17Ø: CALL PR%: PRINT AA$,Ø,5
257Ø  IF SC% THEN AA$ = "E" +  STR$ ( - E2%): CALL PR%: PRINT AA$,28 -  LEN
    (AA$) * 4,11
258Ø DE% = E1%:YS = 45:ST = (M1(1) - W1(1)) / 5: FOR I = W1(1) TO M1(1) STEP
    ST:AA = I
259Ø  ON SC% + 1 GOSUB 14Ø,17Ø: IF SC% AND I = W1(1) THEN AA$ = AA$ + "E"
    +  STR$ ( - E1%)
26ØØ  CALL PR%: PRINT AA$,YS - 4 *  LEN (AA$),186:YS = YS + 47: NEXT I
261Ø AA = M1(1): ON SC% + 1 GOSUB 14Ø,17Ø: CALL PR%: PRINT AA$,252,186
262Ø  IF G% THEN  FOR X = 78 TO 219 STEP 47: FOR Y = 7 TO 182 STEP 5: HPLOT
    X,Y TO X,Y + 3: NEXT : NEXT : FOR Y = 42 TO 15Ø STEP 35: FOR X = 31 TO
    266 STEP 5: HPLOT X,Y TO X + 3,Y: NEXT : NEXT
263Ø  VTAB 2: HTAB 5: PRINT "*   CREAZIONE  DELLO  SFONDO   *"
264Ø  VTAB 1Ø: HTAB 5: FLASH : PRINT "*   CREAZIONE  DEL  GRAFICO    *"
265Ø  NORMAL
266Ø  FOR I = Ø TO NC% - 1
267Ø  VTAB 15: HTAB 5: PRINT "CURVA N.RO "I;: HTAB 2Ø: PRINT "PUNTI :
    "
268Ø  PRINT D$"OPEN "C$(I)DR$
269Ø  PRINT D$"READ "C$(I)
27ØØ  INPUT TR$: CALL CV%:NP = TR$:NP =  ABS (NP)
271Ø EX% = 1: FOR J = 1 TO NP: VTAB 15: HTAB 27: PRINT J
272Ø A% =  PEEK (SI%)
273Ø  INPUT TR$: CALL CV%:X = TR$: INPUT TR$: CALL CV%:Y = TR$
274Ø X = 31 + (X - W1(1)) * FX:Y = 182 - (Y - W2(1)) * FY
275Ø  IF X < 3Ø THEN X = 3Ø: IF FL% = 3 THEN 283Ø
276Ø  IF Y < 6 THEN Y = 6: IF FL% = 3 THEN 283Ø
277Ø  IF X > 267 THEN X = 267: IF FL% = 3 THEN 283Ø
278Ø  IF Y > 183 THEN Y = 183: IF FL% = 3 THEN 283Ø
279Ø  IF FL% = 1 THEN  HPLOT X,Y
28ØØ  IF EX% THEN XV% = X:YV% = Y
281Ø  IF FL% = 3 THEN  CALL PR%: PRINT 'I',X - 1,Y - 1
282Ø  IF FL% = 2 THEN  HPLOT XV%,YV% TO X,Y:XV% = X:YV% = Y
283Ø EX% = Ø
284Ø  NEXT J: PRINT D$"CLOSE "C$(I)
```

APPLE

Seguito listato 1.

```
2850  NEXT I: POKE NO%,Ø
2860  IF AX% * AY% * AS% THEN  HCOLOR= 7: CALL PR%: PRINT '127',AX% + 4,A
    Y% - 6: HCOLOR= Ø: CALL PR%: PRINT "Ø",AX% + 4,AY% - 6
2870  VTAB 1Ø: HTAB 5: PRINT "*   CREAZIONE  DEL  GRAFICO    *"
2880  IF FL% = 3 OR NC% = 1 THEN 291Ø
2890  POKE KC%,Ø: VTAB 2Ø: PRINT  CHR$ (7)"VUOI NUMERARE LE CURVE ? "I$
2900  VTAB 2Ø: HTAB 27: GET Z$: PRINT Z$: IF Z$ <  > "S" AND Z$ <  > "N" THEN
    CALL  - 198: GOTO 29ØØ
2910  POKE SG%,Ø: POKE FS%,Ø: POKE P2%,Ø: POKE HG%,Ø
2920  IF Z$ = "N" OR FL% = 3 OR NC% = 1 THEN 327Ø
2930  FOR I = Ø TO NC% - 1
2940  PRINT D$"OPEN"C$(I)",L8"
2950 R$ = D$ + "READ" + C$(I) + ",R"
2960  PRINT R$Ø
2970  INPUT TR$: CALL CV%:NP = TR$:NN = 1:NP = NP * 2:NP =  ABS (NP)
2980  IF NN < 1 THEN NN = NP - 1
2990  IF NN > NP - 1 THEN NN = 1
3000  PRINT R$NN
3010  INPUT TR$: CALL CV%:X = TR$: INPUT TR$: CALL CV%:Y = TR$
3020 A% =  PEEK (SI%)
3030 CU% = 127: IF X > M1(1) OR X < W1(1) OR Y > M2(1) OR Y < W2(1) THEN
   CU% = 7Ø
3040 PA% = Ø:X = 3Ø + (X - W1(1)) * FX:Y = 179 - (Y - W2(1)) * FY
3050  IF X < 32 THEN X = 32
3060  IF X > 264 THEN X = 264
3070  IF Y < 8 THEN Y = 8
3080  IF Y > 181 THEN Y = 181
3090  POKE KC%,Ø
3100  CALL PR%:X PRINT 'CU%',X,Y:PA% = PA% = Ø: FOR TI = 1 TO TI%: NEXT T
    I
3110 K% =  PEEK (KR%): IF K% < 128 THEN 31ØØ
3120  POKE KC%,Ø
3130  IF PA% THEN  CALL PR%:X PRINT 'CU%',X,Y:PA% = Ø
3140  IF K% = 136 THEN NN = NN - 2: GOTO 298Ø
3150  IF K% = 149 THEN NN = NN + 2: GOTO 298Ø
3160  IF K% = 141 THEN  CALL PR%: PRINT '127',X - 1,Y - 1: CALL PR%: PRINT
    '127',X + 1,Y - 1: CALL PR%: PRINT '127',X - 1,Y + 1: CALL PR%: PRINT
    '127',X + 1,Y + 1: CALL PR%:X PRINT '48 + I',X,Y: GOTO 31ØØ
3170  IF K% = 155 THEN 319Ø
3180  CALL  - 198: GOTO 31ØØ
3190  PRINT D$"CLOSE"C$(I)
3200  POKE KC%,Ø:PA% = Ø: POKE NO%,Ø
3210 A% =  PEEK (SPEAKER%): CALL PR%:X PRINT "OK?",15,22:PA% = PA% = Ø: FOR
   TIME = 1 TO TI%: NEXT TIME
3220 K% =  PEEK (KR%): IF K% <  > 211 AND K% <  > 2Ø6 THEN 321Ø
3230  POKE KC%,Ø: IF PA% THEN PA% = Ø: CALL PR%:X PRINT "OK?",15,22:PA% =
   Ø
3240  IF K% <  > 211 THEN 294Ø
3250  PRINT D$"CLOSE"C$(I)
3260  NEXT I:A% =  PEEK (NO%)
3270  POKE KC%,Ø:PA% = Ø
3280 A% =  PEEK (SPEAKER%): CALL PR%:X PRINT "ESC",15,22:PA% = PA% = Ø: FOR
   TIME = 1 TO TI% * 2: NEXT TIME
3290  IF  PEEK (KR%) <  > 155 THEN 328Ø
3300  IF PA% THEN PA% = Ø: CALL PR%:X PRINT "ESC",15,22:PA% = Ø
3310  TEXT : HOME
3320  VTAB 5: PRINT "   (1)  REGISTRA IL GRAFICO"
3330  VTAB 7: PRINT "   (2)  ALTRE CURVE"
3340  VTAB 9: PRINT "   (3)  CAMBIA LA SCALA"
3350  VTAB 11: PRINT "   (4)  CAMBIA TIPO DI GRAFICO"
3360  VTAB 13: PRINT "   (5)  SCRITTE"
3370  VTAB 15: PRINT "   (6)  FINE"
3380  VTAB 2Ø: PRINT "        QUALE "I$
3390  VTAB 2Ø: HTAB 16: GET Z$: PRINT Z$
3400  IF Z$ = "6" THEN  HOME : END
3410  IF Z$ = "5" THEN 364Ø
3420  IF Z$ = "4" THEN 135Ø
3430  IF Z$ = "3" THEN 35ØØ
3440  IF Z$ = "2" THEN 115Ø
3450  IF Z$ <  > "1" THEN  CALL  - 198: GOTO 339Ø
3460  HOME
3470  VTAB 1Ø: INPUT "NOME DEL GRAFICO ";Z$: IF Z$ = "" THEN 347Ø
3480  PRINT D$"BSAVE"Z$",A$4ØØØ,L$2ØØØ"
3490  GOTO 331Ø
3500  TEXT : HOME
```

Seguito listato 1.

```
3510  GOSUB 170
3520  FOR I = 0 TO NC% - 1
3530  IF W1(0) = XW(I) THEN  INVERSE
3540  VTAB I + 8: HTAB 9:AA = XW(I): GOSUB 140: PRINT AA$: NORMAL
3550  IF M1(0) = XM(I) THEN  INVERSE
3560  VTAB I + 8: HTAB 17:AA = XM(I): GOSUB 140: PRINT AA$: NORMAL
3570  IF W2(0) = YW(I) THEN  INVERSE
3580  VTAB I + 8: HTAB 25:AA = YW(I): GOSUB 140: PRINT AA$: NORMAL
3590  IF M2(0) = YM(I) THEN  INVERSE
3600  VTAB I + 8: HTAB 33:AA = YM(I): GOSUB 140: PRINT AA$: NORMAL
3610  NEXT I
3620  VTAB 20: PRINT "USATI :":I = 12:Z = 1: GOSUB 100
3630  POKE 34,21: GOTO 2170
3640  POKE SG%,0: POKE FS%,0: POKE P2%,0: POKE HG%,0
3650 XP% = 5:YP% = 15:CU% = 77:V% = 20:PA% = 0: POKE KC%,0
3660 A% =  PEEK (SP% * (CU% = 43)) +  PEEK (SP%): CALL PR%:X PRINT 'CU%',
     XP%,YP%:PA% = PA% = 0
3670  FOR TIME = 1 TO TI%: NEXT TIME:K% =  PEEK (KR%): IF K% < 128 THEN 3
     660
3680  POKE KC%,0: IF PA% THEN PA% = 0: CALL PR%:X PRINT 'CU%',XP%,YP%
3690 K% = K% - 128
3700  IF CU% = 43 THEN 4140
3710  IF K% = 27 THEN CO% = CU%:CU% = 43: GOTO 3660
3720  IF (CU% = 115 OR CU% = 120) AND (K% > 64 AND K% < 91) THEN K% = K% +
     32: GOTO 3920
3730  IF CU% = 83 OR CU% = 88 OR CU% = 120 OR CU% = 115 THEN 3920
3740  IF K% = 73 THEN YP% = YP% - V%: GOTO 3790
3750  IF K% = 74 THEN XP% = XP% - V%: GOTO 3790
3760  IF K% = 75 THEN XP% = XP% + V%: GOTO 3790
3770  IF K% = 77 THEN YP% = YP% + V%: GOTO 3790
3780  GOTO 3660
3790  IF YP% < 0 THEN YP% = 0
3800  IF YP% > 187 THEN YP% = 187
3810  IF XP% < 0 THEN XP% = 0
3820  IF XP% > 277 THEN XP% = 277
3830  GOTO 3660
3840 AA$ = ""
3850 A% =  PEEK (SPEAKER%): CALL PR%:X PRINT "?",XP%,YP%:PA% = PA% = 0
3860  FOR TIME = 1 TO TI%: NEXT TI:K% =  PEEK (KR%): IF K% < 128 THEN 385
     0
3870  POKE KC%,0: IF PA% THEN  CALL PR%:X PRINT "?",XP%,YP%:PA% = 0
3875  IF K% = 129 THEN 3900
3880  IF K% > 185 OR K% < 176 THEN  CALL  - 198: GOTO 3850
3890 AA$ = AA$ +  CHR$ (K% - 128): IF  LEN (AA$) < 2 THEN 3850
3900 V% =  VAL (AA$): IF  NOT V% THEN  CALL  - 198: GOTO 3840
3910  GOTO 3660
3920  IF K% <  > 8 THEN 4020
3930  IF B1$ = "" THEN K% = 0: GOTO 3980
3940  IF  LEN (B1$) = 1 THEN K% =  ASC (B1$):B2$ = B2$ + B1$:B1$ = "": GOTO
     3960
3950 K% =  ASC ( RIGHT$ (B1$,1)):B2$ = B2$ +  RIGHT$ (B1$,1):B1$ =  LEFT$
     (B1$, LEN (B1$) - 1)
3960  IF K% = 9 THEN YP% = YP% + 6: GOTO 3790
3970  IF K% = 13 THEN YP% = YP% - 6: GOTO 3790
3980 XP% = XP% - 4: IF XP% < 0 THEN XP% = 277:YP% = YP% - 6
3990  IF YP% < 0 THEN YP% = 0
4000  IF K% THEN  CALL PR%:X PRINT 'K%',XP%,YP%
4010  GOTO 3660
4020  IF K% <  > 21 THEN 4060
4030  IF B2$ = "" THEN K% = 32: GOTO 4110
4040  IF  LEN (B2$) = 1 THEN K% =  ASC (B2$):B2$ = "": GOTO 4070
4050 K% =  ASC ( RIGHT$ (B2$,1)):B2$ =  LEFT$ (B2$, LEN (B2$) - 1)
4060  IF K% = 9 THEN YP% = YP% - 6: GOSUB 340: GOTO 3790
4070  IF K% = 13 THEN YP% = YP% + 6: GOSUB 340: GOTO 3790
4080  IF K% < 32 THEN 3660
4090  IF CU% = 83 OR CU% = 115 THEN  CALL PR%: PRINT 'K%',XP%,YP%
4100  IF CU% = 88 OR CU% = 120 THEN  CALL PR%:X PRINT 'K%',XP%,YP%
4110  GOSUB 340
4120 XP% = XP% + 4: IF XP% > 277 THEN XP% = 0:YP% = YP% + 6
4130  GOTO 3790
4140 CU% = CO%: IF K% = 77 THEN CU% = K%: GOTO 3660
4150  IF K% = 46 THEN CU% = 42:YP% = YP% - 2: GOTO 3790
4160  IF K% = 63 AND CU% = 42 THEN 4390
4170  IF K% = 83 OR K% = 88 THEN CU% = K%: GOTO 3660
4180  IF K% = 86 THEN 3840
```

APPLE

Seguito listato 1.

```
4190  IF K% = 85 AND (CO% = 115 OR CO% = 120) THEN CU% = CO% - 32: GOTO 3
     660
4200  IF K% = 76 AND (CO% = 83 OR CO% = 88) THEN CU% = CO% + 32: GOTO 366
     Ø
4210  IF K% = 81 THEN 3270
4220  IF K% <  > 64 OR CU% = 77 OR CU% = 42 THEN 3660
4230 I = Ø
4240 A% =  PEEK (SPEAKER%): CALL PR%:X PRINT 'I',XP%,YP%:PA% = PA% = Ø
4250  FOR TIME = 1 TO TI%: NEXT TI:K% =  PEEK (KR%): IF K% < 128 THEN 424
     Ø
4260  POKE KC%,Ø: IF PA% THEN  CALL PR%:X PRINT 'I',XP%,YP%:PA% = Ø
4270  IF K% = 141 THEN K% = I: GOTO 4090
4275  IF K% = 155 THEN 3660
4280  IF K% = 136 THEN I = I - 1
4290  IF K% = 149 THEN I = I + 1
4300  IF I < Ø THEN I = 127
4310  IF I > 127 THEN I = Ø
4320  IF I = 122 THEN I = 96
4330  IF I = 97 THEN I = 123
4340  IF I = 94 THEN I = 91
4350  IF I = 92 THEN I = 95
4360  IF I = 90 THEN I = 31
4370  IF I = 32 THEN I = 91
4380  GOTO 4240
4390  IF XP% < 30 OR XP% > 265 OR YP% > 180 OR YP% < 5 THEN  CALL  - 198:
     GOTO 3660
4400 X = (XP% - 30) / FX + W1(1):Y = (180 - YP%) / FY + W2(1)
4410  HOME
4420  VTAB 5: HTAB 5: PRINT "* X = "X
4430  VTAB 8: HTAB 5: PRINT "* Y = "Y
4440  TEXT
4450  VTAB 20: HTAB 5: PRINT "PREMI RETURN "I$
4460  VTAB 20: HTAB 19: GET Z$: IF Z$ <  >  CHR$ (13) THEN 4460
4470  POKE SG%,Ø: POKE FS%,Ø: POKE F2%,Ø: POKE HG%,Ø
4480  GOTO 3660
```

Listato 2 - Trasforma file: trasforma file di dati manipolabili da FP.PLOT.

```
10  HIMEM: - 27368
20 D$ =  CHR$ (4)
30 CV% =  - 27367:TR$ = "NX@@@@F"
40  PRINT D$"BLOAD FP.STORE,A$9519"
50 NO% =  - 16248 +  PEEK (1528):SI% = NO% + 1:I$ =  CHR$ (91) + " ]"
60  DIM X(3000),Y(3000)
70  TEXT : HOME
80  VTAB 5: PRINT " DRIVE FILE ORIGINALE  "I$
90  VTAB 5: HTAB 25: GET Z$: IF Z$ =  CHR$ (13) THEN 110
100 D1$ = Z$
110  PRINT D1$: IF D1$ < "1" AND D1$ <  > "2" THEN  CALL  - 198: GOTO 90
120  VTAB 10: PRINT " DRIVE FILE MODIFICATO "I$
130  VTAB 10: HTAB 25: GET Z$: IF Z$ =  CHR$ (13) THEN 150
140 D2$ = Z$
150  PRINT D2$: IF D2$ <  > "1" AND D2$ <  > "2" THEN  CALL  - 198: GOTO
    130
160  VTAB 20: PRINT "* TUTTO OK ? "I$
170  VTAB 20: HTAB 15: GET Z$: PRINT Z$: IF Z$ <  > "S" AND Z$ <  > "N" THEN
    160
180  IF Z$ = "N" THEN 70
190  HOME
200  VTAB 5: INPUT "FILE ORIGINALE  : ";Z$
210  IF Z$ = "" THEN  VTAB 5: HTAB 19: PRINT F1$: GOTO 230
220 F1$ = Z$
230  IF F1$ = "" THEN  CALL  - 198: GOTO 200
240  VTAB 15: INPUT "FILE MODIFICATO : ";Z$
250  IF Z$ = "" THEN  VTAB 10: HTAB 19: PRINT F2$: GOTO 270
260 F2$ = Z$
270  IF F2$ = "" THEN  CALL  - 198: GOTO 240
280  HOME
290  VTAB 5: PRINT "IL NUMERO PUNTI E' NEL RECORD Ø "I$
300  VTAB 5: HTAB 34: GET Z$: PRINT Z$: IF Z$ <  > "S" AND Z$ <  > "N" THEN
    CALL  - 198: GOTO 300
310  IF Z$ = "S" THEN NN% = Ø: GOTO 350
320  VTAB 8: INPUT " NUMERO PUNTI ? ";Z$: IF Z$ = "" THEN  VTAB 8: HTAB 1
    7: PRINT NN%: GOTO 340
330 NN% =  VAL (Z$)
340  IF NN% <  = Ø THEN  CALL  - 198: GOTO 320
350  HOME : PRINT "* DISPOSIZIONE X E Y NEL FILE ORIGINALE"
360  VTAB 5: HTAB 5: PRINT "(1) X ED Y A COPPIE"
370  VTAB 8: HTAB 5: PRINT "(2) Y ED X A COPPIE"
380  VTAB 11: HTAB 5: PRINT "(3) PRIMA LE X E POI LE Y"
390  VTAB 14: HTAB 5: PRINT "(4) PRIMA LE Y E POI LE X"
400  VTAB 18: HTAB 5: PRINT "QUALE "I$
410  VTAB 18: HTAB 12: GET Z$: PRINT Z$: IF Z$ < "1" OR Z$ > "4" THEN  CALL
     - 198: GOTO 410
420 F% =  VAL (Z$)
430  VTAB 20: HTAB 5: PRINT "TUTTO OK? "I$
440  VTAB 20: HTAB 16: GET Z$: PRINT Z$: IF Z$ <  > "S" AND Z$ <  > "N" THEN
     CALL  - 198: GOTO 440
```

Seguito listato 2.

```
450  IF Z$ = "N" THEN 190
460  HOME : VTAB 10: HTAB 2: PRINT "*    CREAZIONE NUOVO FILE DATI    *"
470  PRINT D$"VERIFY"F1$",D"D1$
480  PRINT D$"OPEN"F1$
490  PRINT D$"READ"F1$
500  IF  NOT NN% THEN  INPUT NN%
510  IF F% < 3 THEN 600
520  FOR L = 1 TO NN%: POKE SI%,Ø
530  INPUT X: IF F% = 3 THEN X(L) = X
540  IF F% = 4 THEN Y(L) = X
550  NEXT
560  FOR L = 1 TO NN%: POKE SI%,Ø
570  INPUT Y: IF F% = 3 THEN Y(L) = Y
580  IF F% = 4 THEN X(L) = Y
590  NEXT : GOTO 650
600  FOR L = 1 TO NN%: POKE SI%,Ø
610  INPUT X: INPUT Y
620  IF F% = 1 THEN X(L) = X:Y(L) = Y
630  IF F% = 2 THEN Y(L) = X:X(L) = Y
640  NEXT
650  PRINT D$"CLOSE"F1$
660  PRINT D$"CLOSE"F1$
670  PRINT D$"OPEN"F2$",D"D2$
680  PRINT D$"WRITE"F2$
690  CALL CV%:TR$ = NN%: PRINT TR$
700  FOR L = 1 TO NN%: POKE SI%,Ø
710  CALL CV%:TR$ = X(L): PRINT TR$
720  CALL CV%:TR$ = Y(L): PRINT TR$
730  NEXT
740  PRINT D$"CLOSE"F2$
750  POKE NO%,Ø
760  HOME
770  VTAB 10: HTAB 5: PRINT "ANCORA ?"I$
780  VTAB 10: HTAB 14: GET Z$: PRINT Z$: IF Z$ <  > "S" AND Z$ <  > "N" THEN
     CALL  - 198: GOTO 780
790  IF Z$ = "S" THEN 70
```

```
*9251.9418

9251- 20 8Ø FE A9 46 85 FD
9258- 20 B1 ØØ AØ ØØ 84 FF B1
9260- B8 C9 58 DØ Ø7 C6 FF 2Ø
9268- B1 ØØ AØ ØØ B1 B8 C9 BA
9270- FØ Ø3 4C Ø5 94 2Ø B1 ØØ
9278- B1 B8 C9 22 FØ 43 C9 27
9280- FØ 6D 2Ø E3 DF A6 11 EØ
9288- FF FØ Ø3 4C ØD 94 85 FA
9290- 84 FB AØ ØØ B1 FA FØ 26
9298- C5 FD 9Ø Ø2 A5 FD 85 FC
92AØ- C8 B1 FA AA C8 B1 FA 85
92A8- FB 86 FA AØ ØØ A6 FD B1
92BØ- FA 29 7F 9D Ø6 92 CA C8
92B8- C4 FC DØ F3 86 FD 4C 18
92CØ- 93 2Ø B1 ØØ AØ ØØ A6 FD
92C8- B1 B8 C9 22 FØ 1Ø EØ ØØ
92DØ- DØ Ø3 4C Ø9 94 29 7F 9D
92D8- Ø6 92 CA C8 DØ EA 86 FD
92EØ- 98 18 65 B8 85 B8 A5 B9
92E8- 69 ØØ 85 B9 4C 15 93 2Ø
92FØ- B1 ØØ 2Ø FF 93 A6 FD A5
92F8- 5Ø 29 7F 9D Ø6 92 C6 FD
93ØØ- FØ Ø8 AØ ØØ B1 B8 C9 2C
93Ø8- FØ E5 AØ ØØ B1 B8 C9 27
931Ø- FØ Ø3 4C Ø9 94 2Ø B1 ØØ
9318- AØ ØØ B1 B8 C9 C8 DØ Ø7
932Ø- A5 FD FØ EE 4C 75 92 C9
9328- 2C FØ Ø3 4C 65 93 2Ø B1
933Ø- ØØ 2Ø FF 93 A5 5Ø C9 18
9338- 8D 4E 92 A5 51 8D 4F 92
934Ø- E9 Ø1 9Ø Ø3 4C 11 94 AØ
9348- ØØ B1 B8 C9 2C FØ Ø3 4C
935Ø- Ø5 94 2Ø B1 ØØ 2Ø FF 93
9358- A5 51 DØ E8 A5 5Ø C9 CØ
936Ø- BØ E2 8D 5Ø 92 A6 FD A9
9368- FF 9D Ø6 92 A9 47 85 FD
937Ø- C6 FD A6 FD BD Ø6 92 C9
9378- FF DØ Ø4 2Ø 84 FE 6Ø ØA
938Ø- AA 65 FC BD 19 94 85 FB
9388- A9 ØØ 85 FA 46 FB 9Ø 1E
939Ø- AE 4E 92 AC 4F 92 AD 5Ø
9398- 92 24 FF 3Ø Ø6 2Ø 57 F4
93AØ- 4C AE 93 2Ø 11 F4 A5 3Ø
93A8- 29 7F 51 26 91 26 A5 FA
93BØ- C9 ØE DØ 1D AD 5Ø 92 38
93B8- E9 Ø4 8D 5Ø 92 18 A9 Ø2
93CØ- 6D 4E 92 8D 4E 92 AD 4F
93C8- 92 69 ØØ 8D 4F 92 4C 7Ø
93DØ- 93 C9 Ø4 DØ 13 AD 5Ø 92
93D8- 38 E9 Ø4 8D 5Ø 92 EE 4E
93EØ- 92 DØ 18 EE 4F 92 DØ 13
93E8- C9 Ø9 FØ E9 C9 Ø7 DØ Ø8
93FØ- A6 FC E8 BD 19 94 85 FB
93F8- EE 5Ø 92 E6 FA DØ 8D 2Ø
94ØØ- 67 DD 4C 52 E7 A2 1Ø DØ
94Ø8- ØA A2 4D DØ Ø6 A2 A3 DØ
941Ø- Ø2 A2 35 2Ø 8Ø FE 4C 12
9418- D4

*9519.95FF

9519- 2Ø 8Ø FE 2Ø B1 ØØ 2Ø
952Ø- E3 DF A6 12 FØ Ø3 4C F7
9528- 95 A6 11 DØ 58 85 FB 84
953Ø- FC AØ ØØ B1 B8 C9 DØ FØ
9538- Ø3 4C F3 95 2Ø B1 ØØ 2Ø
954Ø- E3 DF A6 12 FØ Ø3 4C F7
9548- 95 A6 11 FØ F9 85 F9 84
955Ø- FA 2Ø DC 95 AØ Ø5 B1 F9
9558- 85 FD C8 B1 F9 85 FE AØ
956Ø- ØØ B1 F9 ØA ØA 46 FE 6A
9568- 46 FE 6A 91 FB C8 CØ Ø2
957Ø- DØ EF B1 F9 ØA ØA 46 FD
9578- 6A 46 FD 6A 91 FB C8 CØ
958Ø- Ø5 DØ EF FØ 43 85 F9 84
9588- FA 2Ø DC 95 AØ ØØ B1 B8
959Ø- C9 DØ DØ A5 2Ø B1 ØØ 2Ø
9598- 67 DD A5 9E 24 A2 3Ø Ø2
95AØ- 29 7F 85 9E AØ Ø4 B9 9D
95A8- ØØ Ø9 4Ø 29 7F 91 F9 88
95BØ- CØ FF DØ F2 A2 Ø4 16 9D
95B8- 2A 16 9D 2A CA EØ FF DØ
95CØ- ØB C8 Ø9 4Ø 29 7F 91 F9
95C8- 2Ø 84 FE 6Ø EØ Ø1 DØ E6
95DØ- Ø9 4Ø 29 7F AØ Ø5 91 F9
95D8- A9 ØØ FØ DA AØ ØØ B1 F9
95EØ- C9 Ø7 FØ Ø3 4C FB 95 C8
95E8- B1 F9 AA C8 B1 F9 86 F9
95FØ- 85 FA 6Ø A2 1Ø DØ Ø6 A2
95F8- A3 DØ Ø2 A2 BØ 4C 12 D4
*
```

Listato 3 - Codici di PRINT 3 ★ 5 ($9251.9418), del SET DI CARATTERI e di FP.STORE.

Seguito listato 3.

```
*9419.9518

9419- A7 1C E2 08 45 14 05
9420- 14 A2 08 E5 14 47 1C E7
9428- 1C 07 1C A5 14 08 22 80
9430- 00 E2 0B E8 23 B9 78 AF
9438- 4A 1E 19 BB 6E AE 56 B5
9440- 3A 0F 3E 3E 78 4A 11 44
9448- 29 A4 12 A9 3A 35 55 EA
9450- 2B BB 56 F2 4A AE 3A 9F
9458- 3C 00 00 E0 02 03 0C 5F
9460- 7D F2 27 99 4C 6C 32 60
9468- 00 2E 02 20 3A BA 28 C4
9470- 11 10 01 84 10 00 02 98
9478- 0C 2E 3A F2 43 BA 4A B5
9480- 2A 0F 71 B7 26 AE 26 B9
9488- 0C AA 2A B2 3A 40 01 50
9490- 01 44 45 4A 29 51 11 A1
9498- 0A 3F 5A BE 78 BF 2A 2E
94A0- 46 3F 3A BF 46 BF 04 2E
94A8- 76 9F 7C F1 47 08 3E 9F
94B0- 6C 1F 42 5F 7C 3F 78 3F
94B8- 7E BF 08 3F 7F BF 68 B2
94C0- 26 E1 07 1F 7E 0F 3E 9F
94C8- 7D 9B 6C 83 0F B9 4E 3F
94D0- 46 83 60 31 7E 44 10 10
94D8- 42 41 10 88 7A 9F 22 88
94E0- 52 88 7E 5C 5B BE 04 54
94E8- 7B 9F 60 A4 03 08 36 1F
94F0- 51 F1 43 5E 78 5E 70 4C
94F8- 32 5E 11 44 79 5E 10 D6
9500- 6A 4F 22 0E 7A 0E 3A 1E
9508- 79 92 49 96 7A DA 4A C4
9510- 45 E0 03 D1 11 22 08 FF
9518- 7F
```

Listato 4 - Utility per memorizzare le variabili FP come stringhe di sette caratteri.

```
!ASM
**END OF PASS 1
**END OF PASS 2
0800             1  ******************************
0800             2  * UTILITY PER MEMORIZZARE LE *
0800             3  * VARIABILI FP COME STRINGHE *
0800             4  * DI 7 CARATTERI (by adriano)*
0800             5  ******************************
0800             6  ;
9519             7          ORG $9519
9519             8          OBJ $800
00F9             9  PS.L    EPZ $F9
00FA            10  PS.H    EPZ $FA          ;PUNTATORI VAR $
00FB            11  PN.L    EPZ $FB
00FC            12  PN.H    EPZ $FC          ;PUNTATORI VAR FP
00FD            13  BY.1    EPZ $FD
00FE            14  BY.2    EPZ $FE
009D            15  FP      EPZ $9D          ;FLOATING POINT ACCUMULATOR
9519 2080FE     16          JSR $FE80        ;INVERSE
951C            17  ; I PASSI 19-25 DECIDONO SE LA CONVERSIONE E'
951C            18  ;DA STRINGA A FP O VICEVERSA
951C 20B100     19          JSR $B1          ;PUNTA LA VARIABILE
951F 20E3DF     20          JSR $DFE3        ;LA TROVA
9522 A612       21          LDX $12
9524 F003       22          BEQ OK           ;NON E' UNA VAR %
9526 4CF795     23          JMP TYPE         ; VAR DI TIPO ERRATO
9529 A611       24  OK      LDX $11
952B D058       25          BNE STRINGA
952D            26  ;I PASSI 27-73 CONVERTONO L'ACCUMUL. FP IN STRINGA
952D 85FB       27          STA PN.L         ;PUNTATORI DELLA
952F 84FC       28          STY PN.H         ;VAR NUMERICA
9531 A000       29          LDY #0
9533 B1B8       30          LDA ($B8),Y      ;PROSSIMO CARATTERE
9535 C9D0       31          CMP #!208        ;E' UN UGUALE?
9537 F003       32          BEQ OK2
9539 4CF395     33  SYN     JMP SYNTAX       ;NO! ERRORE
953C 20B100     34  OK2     JSR $B1          ;PUNTA L'ALTRA VAR
953F 20E3DF     35          JSR $DFE3        ;E LA TROVA
9542 A612       36          LDX $12          ;E' UNA % ?
9544 F003       37          BEQ OK3
9546 4CF795     38  ERR1    JMP TYPE         ;SI! ERRORE
9549 A611       39  OK3     LDX $11
954B F0F9       40          BEQ ERR1
954D 85F9       41          STA PS.L         ;PUNTATORI DELLA
954F 84FA       42          STY PS.H         ;STRINGA
9551 20DC95     43          JSR CTRL$
9554 A005       44          LDY #5
9556 B1F9       45          LDA (PS.L),Y
9558 85FD       46          STA BY.1
955A C8         47          INY
955B B1F9       48          LDA (PS.L),Y
955D 85FE       49          STA BY.2
955F A000       50          LDY #0
9561 B1F9       51  LOOPA   LDA (PS.L),Y
9563 0A         52          ASL
9564 0A         53          ASL
9565 46FE       54          LSR BY.2
9567 6A         55          ROR
```

Seguito listato 4.

```
9568 46FE      56            LSR BY.2
956A 6A        57            ROR
956B 91FB      58            STA (PN.L),Y
956D C8        59            INY
956E C002      60            CPY #2
9570 D0EF      61            BNE LOOPA
9572 B1F9      62  LOOPB     LDA (PS.L),Y
9574 0A        63            ASL
9575 0A        64            ASL
9576 46FD      65            LSR BY.1
9578 6A        66            ROR
9579 46FD      67            LSR BY.1
957B 6A        68            ROR
957C 91FB      69            STA (PN.L),Y
957E C8        70            INY
957F C005      71            CPY #5
9581 D0EF      72            BNE LOOPB
9583 F043      73            BEQ RTS
9585           74  ;I PASSI 75-118 CONVERTONO LA STRINGA IN FP
9585 85F9      75  STRINGA   STA PS.L
9587 84FA      76            STY PS.H
9589 20DC95    77            JSR CTRL$
958C A000      78            LDY #0
958E B1B8      79            LDA ($B8),Y
9590 C9D0      80            CMP #208
9592 D0A5      81            BNE SYN
9594 20B100    82            JSR $B1              ;PUNTA IL NUMERO
9597 2067DD    83            JSR $DD67            ;E LO TROVA
959A A59E      84            LDA $9E              ;METTE A POSTO IL SEGNO
959C 24A2      85            BIT $A2
959E 3002      86            BMI NEGATIVO
95A0 297F      87            AND #$7F             ;E' POSITIVO
95A2 859E      88  NEGATIVO  STA $9E
95A4 A004      89            LDY #4
95A6 B99D00    90  LOOPC     LDA FP,Y
95A9 0940      91            ORA #$40             ; ALZA IL BIT 6
95AB 297F      92            AND #$7F
95AD 91F9      93            STA (PS.L),Y
95AF 88        94            DEY
95B0 C0FF      95            CPY #$FF
95B2 D0F2      96            BNE LOOPC
95B4 A204      97            LDX #4
95B6 169D      98  LOOPD     ASL FP,X
95B8 2A        99            ROL
95B9 169D     100            ASL FP,X
95BB 2A       101            ROL
95BC CA       102            DEX
95BD E0FF     103            CPX #$FF
95BF D00B     104            BNE CONT1
95C1 C8       105            INY
95C2 0940     106            ORA #$40
95C4 297F     107            AND #$7F
95C6 91F9     108            STA (PS.L),Y
95C8 2084FE   109  RTS       JSR $FE84            ;NORMAL
95CB 60       110            RTS
95CC E001     111  CONT1     CPX #1
95CE D0E6     112            BNE LOOPD
95D0 0940     113            ORA #$40
95D2 297F     114            AND #$7F
95D4 A005     115            LDY #5
95D6 91F9     116            STA (PS.L),Y
95D8 A900     117            LDA #0
95DA F0DA     118            BEQ LOOPD
95DC          119  ;I PASSI 121-133 CONTROLLANO CHE LA STRINGA SIA
95DC          120  ;LUNGA 7 CARATTERI E NE SALVANO I PUNTATORI
95DC A000     121  CTRL$     LDY #0
95DE B1F9     122            LDA (PS.L),Y         ;LUNGHEZZA $
95E0 C907     123            CMP #7
95E2 F003     124            BEQ OK4
95E4 4CFB95   125            JMP ERR$             ;LUNGHEZZA ERRATA
95E7 C8       126  OK4       INY
95E8 B1F9     127            LDA (PS.L),Y
95EA AA       128            TAX
95EB C8       129            INY
95EC B1F9     130            LDA (PS.L),Y
95EE 86F9     131            STX PS.L             ;I "VERI" PUNTATORI
95F0 85FA     132            STA PS.H             ;DELLA STRINGA
95F2 60       133            RTS
```

APPLE

Seguito listato 4.

```
95F3            134  ;I PASSI 136-140 DEFINISCONO IL TIPO DI ERRORE
95F3            135  ;E SALTANO AD UNA ROUTINE BASIC DI GESTIONE D'ERRORE
95F3 A210       136  SYNTAX   LDX #!16
95F5 D006       137           BNE ERRORE
95F7 A2A3       138  TYPE     LDX #!163
95F9 D002       139           BNE ERRORE
95FB A2B0       140  ERR$     LDX #!176
95FD 4C12D4     141  ERRORE   JMP $D412
                142           END
***** END OF ASSEMBLY
```

Listato 5 - Listato dell'utility per scrivere sulle pagine grafiche dell'Apple.

```
!ASM
**END OF PASS 1
**END OF PASS 2
0800              1  ******************************
0800              2  * UTILITY PER SCRIVERE SULLE *
0800              3  * PAGINE GRAFICHE DELL'APPLE *
0800              4  * CON CARATTERI DI 3*7 PUNTI *
0800              5  * (by adriano baracco - '83) *
0800              6  ******************************
0800              7  ;
9206              8           ORG $9206
9206              9           OBJ $800
9251             10  BUFFER   DFS !75              ;BUFFER CARATTERI+VARIABILI
924E             11  X.L      EQU BUFFER+!72
924F             12  X.H      EQU X.L+1
9250             13  Y        EQU X.H+1
00FF             14  XFLAG    EPZ $FF
00FA             15  PUNT.L   EPZ $FA
00FB             16  PUNT.H   EPZ $FB
00FA             17  COUNTER  EPZ PUNT.L
00FB             18  CAR      EPZ PUNT.H
00FC             19  LUNGH    EPZ $FC
00FC             20  PUNTAT   EPZ LUNGH
00FD             21  CONT.X   EPZ $FD
9251 2080FE      22           JSR $FE80             ; INVERSE
9254 A946        23           LDA #!70
9256 85FD        24           STA CONT.X            ; E' IL CONTATORE PER IL BUFFER
9258 20B100      25           JSR $B1               ; PUNTA AL COMANDO
925B A000        26           LDY #0
925D 84FF        27           STY XFLAG             ; AZZERA IL FLAG DI XPRINT
925F B1B8        28           LDA ($B8),Y
9261 C958        29           CMP #'X'              ; E' XPRINT ?
9263 D007        30           BNE PRINT
9265 C6FF        31           DEC XFLAG             ; SI' : FLAG=$FF
LISA 2.2
9267 20B100      32           JSR $B1
926A A000        33           LDY #0
926C B1B8        34  PRINT    LDA ($B8),Y
926E C9BA        35           CMP #!186             ; E' PRINT?
9270 F003        36           BEQ ANCORA
9272 4C0594      37           JMP SYNTAX            ; NO! ERRORE
9275 20B100      38  ANCORA   JSR $B1
9278 B1B8        39           LDA ($B8),Y
927A C922        40           CMP #'"'              ; E' UN MESSAGGIO
927C F043        41           BEQ MESSAG            ; SI': MEMORIZZALO
927E C927        42           CMP #'''              ; SONO CODICI ASC
9280 F06D        43           BEQ ASC               ; SI' : MEMORIZZALI
9282 20E3DF      44           JSR $DFE3             ; E' UNA VARIABILE
9285 A611        45           LDX $11
9287 E0FF        46           CPX #$FF              ; E' UNA VARIABILE ALFANUMERICA ?
9289 F003        47           BEQ PUNTA
928B 4C0D94      48           JMP TYPE              ;NO! ERRORE
928E             49  ;I PASSI DA 49 A 75 MEMORIZZANO LA STRINGA NEL BUFFER
928E 85FA        50  PUNTA    STA PUNT.L            ; MEMORIZZA I PUNTATORI
9290 85FB        51           STA PUNT.H            ; DELLA STRINGA
9292 A000        52           LDY #0
9294 B1FA        53           LDA (PUNT.L),Y        ; LUNGHEZZA DELLA STRINGA
9296 F026        54           BEQ JMPX.Y
9298 C5FD        55           CMP CONT.X            ; PIU' LUNGA DEL BUFFER?
929A 9002        56           BCC MEMOVAR           ; NO! SALVA TUTTA LA STRINGA
929C A5FD        57           LDA CONT.X            ; SI' SALVA I CARATTERI CHE PUOI
929E 85FC        58  MEMOVAR  STA LUNGH
92A0 C8          59           INY
92A1 B1FA        60           LDA (PUNT.L),Y        ; INIZIO STRINGA, BYTE BASSO
92A3 AA          61           TAX
92A4 C8          62           INY
```

Seguito listato 5.

```
92A5 B1FA     63            LDA (PUNT.L),Y      ; INIZIO STRINGA, BYTE ALTO
92A7 85FB     64            STA PUNT.H
92A9 86FA     65            STX PUNT.L
92AB A000     66            LDY #0
92AD A6FD     67            LDX CONT.X
92AF B1FA     68   LOOP1    LDA (PUNT.L),Y
92B1 297F     69            AND #$7F
92B3 9D0692   70            STA BUFFER,X
92B6 CA       71            DEX
92B7 C8       72            INY
92B8 C4FC     73            CPY LUNGH
92BA D0F3     74            BNE LOOP1
92BC 86FD     75            STX CONT.X
92BE 4C1893   76   JMPX.Y   JMP X.Y.2
92C1          77   ; I PASSI DA 77 A 99 MEMORIZZANO IL MESSAGGIO NEL BUFFER
92C1 20B100   78   MESSAG   JSR $B1             ; PUNTA INIZIO MESSAGGIO
92C4 A000     79            LDY #$0
92C6 A6FD     80            LDX CONT.X
92C8 B1B8     81   LOOP2    LDA ($B8),Y
92CA C922     82            CMP #'"'            ; MESSAGGIO FINITO ?
92CC F010     83            BEQ ENDMESS         ; SI': ESCI
92CE E000     84            CPX #0              ; FINITO IL BUFFER
92D0 D003     85            BNE MEMO            ; C'E ANCORA SPAZIO
92D2 4C0994   86            JMP MEMORY
92D5 297F     87   MEMO     AND #$7F
92D7 9D0692   88            STA BUFFER,X
92DA CA       89            DEX
92DB C8       90            INY
92DC D0EA     91            BNE LOOP2           ; CONTINUA
92DE 86FD     92   ENDMESS  STX CONT.X
92E0 98       93            TYA
92E1 18       94            CLC
92E2 65B8     95            ADC $B8             ; AGGIORNA IL PUNTATORE
92E4 85B8     96            STA $B8
92E6 A5B9     97            LDA $B9
92E8 6900     98            ADC #0
92EA 85B9     99            STA $B9
92EC 4C1593   100           JMP X.Y
92EF          101  ; I PASSI DA 101 A 117 MEMORIZZANO I CODICI ASC
92EF 20B100   102  ASC      JSR $B1             ; PUNTA AL NUOVO VALORE
92F2 20FF93   103           JSR DECO            ; CALCOLA L'ASC
92F5 A6FD     104           LDX CONT.X
92F7 A550     105           LDA $50             ; E' L'ASC
92F9 297F     106           AND #$7F
92FB 9D0692   107           STA BUFFER,X
92FE C6FD     108           DEC CONT.X          ; C'E ANCORA SPAZIO
9300 F008     109           BEQ CTRL'           ; NO! CONTROLLA CHE SIA FINITO
9302 A000     110           LDY #0
9304 B1B8     111           LDA ($B8),Y
9306 C92C     112           CMP #','            ; ALTRI VALORI DA TRADURRE?
9308 F0E5     113           BEQ ASC             ; SI': CONTINUA
930A A000     114  CTRL'    LDY #0
930C B1B8     115           LDA ($B8),Y
930E C927     116           CMP #'''            ; FINITO?
9310 F003     117           BEQ X.Y
9312 4C0994   118  ERRM     JMP MEMORY          ; NO! ERRORE
9315          119  ;I PASSI DA 119 A 153 CONTROLLANO SE VI E' ALTRO DA SCRIVERE
9315          120  ;E TROVANO (SE SPECIFICATE) LE COORDINATE DI X E DI Y
9315 20B100   121  X.Y      JSR $B1             ; AGGIORNA IL PUNTATORE
9318 A000     122  X.Y.2    LDY #0
931A B1B8     123           LDA ($B8),Y
931C C9C8     124           CMP #!200           ; ALTRO DA SCRIVERE?
931E D007     125           BNE TROVAX
9320 A5FD     126           LDA CONT.X          ;SI'! CONTROLLA IL BUFFER
9322 F0EE     127           BEQ ERRM
9324 4C7592   128           JMP ANCORA          ; C'E' SPAZIO: CONTINUA
9327 C92C     129  TROVAX   CMP #','            ;C'E' UNA VIRGOLA ?
9329 F003     130           BEQ TROVAX2
932B 4C6593   131           JMP STAMPA
932E 20B100   132  TROVAX2  JSR $B1
9331 20FF93   133           JSR DECO            ; TROVA IL VALORE DI X
9334 A550     134           LDA $50             ; E' IL BYTE BASSO DELLA X
9336 C918     135           CMP #!24
9338 8D4E92   136           STA X.L
933B A551     137           LDA $51             ; E' IL BYTE ALTO
933D 8D4F92   138           STA X.H
9340 E901     139           SBC #1
9342 9003     140           BCC X<280
```

APPLE

Seguito listato 5.

```
9344 4C1194   141  ERRQ     JMP QUANTITY         ;X>279!
9347 A000     142  X<280    LDY #0
9349 B1B8     143           LDA ($B8),Y
934B C92C     144           CMP #','             ; C'E' UNA VIRGOLA DOPO LA X ?
934D F003     145           BEQ TROVAY
934F 4C0594   146           JMP SYNTAX
9352 20B100   147  TROVAY   JSR $B1              ; PUNTA ALLA Y
LISA 2.2
9355 20FF93   148           JSR DECO             ; TROVA LA Y
9358 A551     149           LDA $51
935A D0E8     150           BNE ERRQ             ; Y>255
935C A550     151           LDA $50              ; E' LA Y
935E C9C0     152           CMP #!192
9360 B0E2     153           BCS ERRQ             ; Y>=192!
9362 8D5092   154           STA Y
9365          155  ;I PASSI DA 154 A 222 SCRIVONO IL CONTENUTO DEL BUFFER
9365 A6FD     156  STAMPA   LDX CONT.X
9367 A9FF     157           LDA #$FF
9369 9D0692   158           STA BUFFER,X
936C A947     159           LDA #!71
936E 85FD     160           STA CONT.X
9370 C6FD     161  LOOP     DEC CONT.X
9372 A6FD     162           LDX CONT.X
9374 BD0692   163           LDA BUFFER,X         ; LEGGE IL CARATTÈRE
9377 C9FF     164           CMP #$FF             ; FINE?
9379 D004     165           BNE CONT             ; NO: CONTINUA
937B 2084FE   166           JSR $FE84
937E 60       167           RTS
937F 0A       168  CONT     ASL                  ; MOLTIPLICA PER DUE
9380 AA       169           TAX                  ; PUNTA NELLA TABELLA
9381 85FC     170           STA PUNTAT
9383 BD1994   171           LDA TAB,X
9386 85FB     172           STA CAR
9388 A900     173           LDA #00
938A 85FA     174           STA COUNTER
938C 46FB     175  LSR      LSR CAR
938E 901E     176           BCC AVANTI           ; NON DISEGNA IL PUNTO
9390 AE4E92   177           LDX X.L
9393 AC4F92   178           LDY X.H
9396 AD5092   179           LDA Y
9399 24FF     180           BIT XFLAG            ; E' XPLOT?
939B 3006     181           BMI XPLOT            ; SI!
939D 2057F4   182           JSR $F457            ; PLOT
93A0 4CAE93   183           JMP AVANTI
93A3 2011F4   184  XPLOT    JSR $F411
93A6 A530     185           LDA $30
93A8 297F     186           AND #$7F             ; NON MODIFICA IL COLORE
93AA 5126     187    .      EOR ($26),Y
93AC 9126     188           STA ($26),Y
93AE A5FA     189  AVANTI   LDA COUNTER
93B0 C90E     190           CMP #$E
93B2 D01D     191           BNE AV1
93B4 AD5092   192           LDA Y                ; FINITO!
93B7 38       193           SEC
93B8 E904     194           SBC #4
93BA 8D5092   195           STA Y
93BD 18       196           CLC
93BE A902     197           LDA #2
93C0 6D4E92   198           ADC X.L
93C3 8D4E92   199           STA X.L
93C6 AD4F92   200           LDA X.H
93C9 6900     201           ADC #0
93CB 8D4F92   202           STA X.H
93CE 4C7093   203           JMP LOOP
93D1 C904     204  AV1      CMP #4
93D3 D013     205           BNE AV2
93D5 AD5092   206  DECY     LDA Y                ; FINITA UNA COLONNA DI PUNTI
93D8 38       207           SEC
93D9 E904     208           SBC #4
93DB 8D5092   209           STA Y
93DE EE4E92   210           INC X.L
93E1 D018     211           BNE INCCON
93E3 EE4F92   212           INC X.H
93E6 D013     213           BNE INCCON
93E8 C909     214  AV2      CMP #9
93EA F0E9     215           BEQ DECY
93EC C907     216           CMP #7
```

Seguito listato 5.

```
93EE D008    217             BNE INCY
93F0 A6FC    218             LDX PUNTAT          ; SECONDO BYTE DEL CARATTERE.
93F2 E8      219             INX
93F3 BD1994  220             LDA TAB,X
93F6 85FB    221             STA CAR
93F8 EE5092  222   INCY      INC Y
93FB E6FA    223   INCCON    INC COUNTER
93FD D08D    224             BNE LSR
93FF 2067DD  225   DECC      JSR $DD67           ; INTERPRETA LE ESPRESSIONI
9402 4C52E7  226             JMP $E752           ; TRADUCE IN ESADECIMALE
9405         227   ;I PASSI DA 226 A 235 DEFINISCONO IL TIPO D'ERRORE
9405         228   ;E SALTANO AD UNA ROUTINE BASIC DI GESTIONE ERRORE
9405 A210    229   SYNTAX    LDX #!16
9407 D00A    230             BNE ERRORE
9409 A24D    231   MEMORY    LDX #!77
940B D006    232             BNE ERRORE
940D A2A3    233   TYPE      LDX #!163
940F D002    234             BNE ERRORE
9411 A235    235   QUANTITY  LDX #!53
9413 2080FE  236   ERRORE    JSR $FE80
9416 4C12D4  237   END       JMP $D412
9419         238   TAB       EQU END+3           ;TABELLA DI DEFINIZIONE CARATTERI
             239             END
***** END OF ASSEMBLY
!PR#0
```

TI 99/4A

Rubrica personale

di **Sergio Borsani**

Per quale motivo tante persone usano un piccolo computer domestico? Le statistiche dicono che quasi il 70 per cento di queste lo usano per i video giochi senza meglio specificare a cosa è dedito l'altro eletto 30 per cento.
Ma c'è forse chi usa un home computer per uno scopo davvero utile, che faccia risparmiare tempo e fatica in qualche applicazione specifica? Non mancheranno le eccezioni, ma penso che nella maggior parte dei casi non si possa parlare di vera utilità e che si tratti piuttosto di un gioco intellettuale, una sorta di jogging per la mente, spesso con apprezzabili contenuti didattici. Facciamo un esempio concreto. Non manca per ogni piccolo computer un programma per la raccolta degli indirizzi che tra l'altro potrebbe diventare un protagonista in clima di vacanze natalizie quando ci si accinge a spedire i fatidici biglietti di auguri. Ma sinceramente non è più pratica e veloce la tradizionale ricerca su una comunissima agenda?

Ciò nonostante ci accingiamo a presentare proprio un programma di questo tipo semplicemente a scopo didattico e quasi accogliendo il pretesto per parlare di alcune caratteristiche del TI994A home computer e del TI Extended BASIC.
È noto come l'uso di un home computer in configurazione base, con un solo registratore quale memoria di massa, sia alquanto limitante per la gestione dei file. Oltre ad una maggiore lentezza, non è possibile avere un accesso diretto ai record che pertanto devono essere letti sequenzialmente. Per ovviare a questo inconveniente, se il file non è molto ampio, esso si può caricare totalmente nella memoria centrale, dopo di che su di esso si potranno svolgere tutte quelle operazioni che normalmente si effettuano sui file su disco, come la ricerca con un campo chiave, la modifica di un record o l'aggiunta di nuovi record.
Iniziamo ad esaminare il listato. Alla linea 320 c'è il menu con cinque opzioni (vedi figura 1).
Ogni record è costituito da cinque campi: cognome e nome, indirizzo, città, numero telefonico, annotazioni varie; con poche variazioni non sostanziali è tuttavia possibile adattare il programma ad altri tracciati record.

Nella RAM l'intero file è contenuto sotto forma di matrice a due dimensioni: R$. Il dimensionamento si limita a cento elementi. L'inserimento dei dati non avviene con istruzioni INPUT ma, in modo più elegante, per mezzo di una cosiddetta "maschera"; cioè ogni campo è introdotto con un'istruzione ACCEPT AT sempre in una posizione del video dove viene mostrata la lunghezza massima del campo e dove una legenda ne chiarifica il contenuto. La maschera viene creata con la subroutine 1890. I campi definiti con l'istruzione

```
1. SCRIVI RECORD
2. AGGIUNGI RECORD
3. LEGGI RECORD
4. REGISTRA FILE
5. CARICA FILE
6. FINE PROGRAMMA
```

Figura 1 - Il menu di inizio.

TRACCIATO RECORD

CAMPO 1 cogn. e nome	CAMPO 2 indirizzo	CAMPO 3 c.a.p. città	CAMPO 4 n. tel.	CAMPO 5 note
1 – 28	29 – 56	57 – 84	85 – 98	99 – 126

Figura 2 - La struttura record.

Figura 3 - Esempio di matrice bidimensionale creata per contenere i dati.

Figura 4 - Stampa del record.

450 vengono portati ad una lunghezza prefissata con un'altra subroutine, la 2010, che aggiunge un opportuno numero di space. I record così creati non vengono subito registrati su nastro ma sono trattenuti in memoria centrale per poterli controllare ed eventualmente correggere (opzione 3: leggi record).

Poiché con le registrazioni su nastro non è a disposizione la funzione EOF (End Of File), è necessario riempire un campo dell'ultimo record con una parola chiave. Con notevole fantasia è stata scelta la parola "FINEFILE"! (linea 700).

Il flag denominato FLAGFILE (linea 730) è impostato a 1 o dopo la scrittura del file o dopo il suo caricamento dal nastro e garantisce la presenza del file al momento della sua lettura. Un contatore tiene nota dei record appena scritti o caricati da nastro; al termine di queste operazioni il numero totale dei record viene conservato nella variabile TOTREC (linea 770). Alla linea successiva inizia la sezione per la lettura del file e per eventuali modifiche; inoltre è prevista una lettura sequenziale ed una lettura diretta. Per la prima, naturalmente, non sarebbe stato necessario caricare l'intero file in memoria; questa operazione è tuttavia indispensabile per simulare un file relative, ad accesso diretto. In questo caso la ricerca può essere fatta attraverso un campo chiave che può essere uno qualsiasi di quelli precedentemente citati ad eccezione dell'ultimo riservato ad annotazioni varie.

Normalmente in questi casi si crea un file Indice contenente un campo chiave ed il relativo numero di record; trovato il nominativo interessato, si va a leggere l'intero record tramite il numero che lo contraddistingue.

Nel nostro caso siamo fortunati perché, in un certo senso, tutti i campi sono strutturati in un file Indice, basterà leggere sequenzialmente gli elementi di una colonna della matrice R$ e, una volta trovato l'elemento che interessa, leggere tutta la riga corrispondente. Nel programma la ricerca è svolta dal ciclo FOR NEXT di linea 1360 e se questa ha buon esito si passa alla linea 1560 per la "videata" dell'intero record.

Il programma non è poi così semplice come potrebbe sembrare a prima vista; un aneddoto chiarirà meglio dove stanno le difficoltà. Un giorno si doveva telefonare all'editore Muzzio di Padova e, trovandosi ad un posto pubblico senza l'agenda personale, ci si dovette rivolgere alla SIP per avere il numero telefonico. La ricerca viene svolta ora in modo computerizzato; nonostante ciò l'impiegata non riusciva a trovare il nominativo richiesto ed il motivo, lo scoprii in seguito, era semplicissimo: la ragione sociale della casa editrice era Franco Muzzio e non Muzzio Franco.

Per ovviare a questi inconvenienti è stato reso il programma più elastico e la ricerca può essere indifferentemente effettuata tramite cognome e nome oppure soltanto attraverso il cognome; il nome della città può essere individuato anche se nel record è preceduto dal CAP e, analogamente, il numero telefonico potrà essere preceduto, oppure no, dal prefisso; è sufficiente che questi elementi siano separati, nel campo chiave, da uno spazio, da una barra obliqua o da una parentesi tonda. È previsto anche il trattamento degli omonimi, pertanto il programma è in grado di estrarre dal file tutti i record di persone residenti, ad esempio, a Milano.

Vedete, alla linea 1000 e seguenti, lo statement CALL EDITOR? Non cercate questa istruzione nel manuale dell'Extended BASIC, non la trovereste. Essa è la chiamata di un sottoprogramma, che è stato chiamato EDITOR perché svolge la funzione di editor di linea. I vantaggi di un sottoprogramma rispetto ad una subroutine, richiamata con una GOSUB, è che il primo ha le variabili indipendenti dalle altre del programma, anche se hanno lo stesso nome! Un sottoprogramma può quindi venire scritto e controllato come un programma indipendente per poi essere agganciato ad un altro programma con un comando MERGE (per altro disponibile solo con un sistema a dischi).

Figura 5 - Il diagramma a blocchi.

Gli elementi di sintassi di una chiamata di un sottoprogramma sono:

CALL nome (parametro 1, parametro 2,...)

....

END - fine del programma principale -

SUB nome (parametro 1, parametro 2,...)

... - sottoprogramma -

SUBEND - fine del sottoprogramma -

Il sottoprogramma deve essere posto al termine del programma principale e tra essi è possibile trasferire variabili o costanti tramite una lista di parametri che accompagna la CALL.

Si parla molto del Pascal e della programmazione strutturata, ma con l'uso dei sottoprogrammi, consentito dal TI Extended BASIC, si possono ottenere ugualmente procedure e strutturazione!

La funzione dell'editor di linea è quella di modificare o correggere un record senza doverlo riscrivere completamente. Quando in risposta alla domanda: "modifica? (y/n)", si preme "Y", compare un cursore sul primo carattere del primo campo del record che può essere normalmente spostato usando il tasto FCTN ed uno di quelli con la freccia: "S" o "D". Premendo ENTER il cursore si sposta sui campi successivi ed una volta portato nella giusta posizione si può scrivere modificando il contenuto di un campo.

I nuovi caratteri digitati andranno a sostituire i precedenti; non sono invece disponibili le funzio-

ni DELETE ed INSERT. Pensate! Con il TI99/4A potete normalmente editare una stringa che sia stata stampata sullo schermo con una PRINT o con una DISPLAY AT?

Un'altra funzione del programma è quella di aggiungere record ad un file già esistente. Per chi possiede un sistema a dischi la cosa non presenta difficoltà, è sufficiente aprire il file nel modo APPEND. L'unica alternativa per chi usa un registratore è quella di caricare in memoria tutto il file, aggiungere i record, quindi registrare nuovamente tutto il file. È questa la soluzione qui adottata. Durante il caricamento del file la variabile CTR (ConTatoRe) conta i record presenti, il numero massimo viene posto in TOTREC, quindi si scrivono altri record a partire dal numero successivo.

Al momento di salvare il file i campi di ogni record vengono sommati a formare un'unica stringa di 126 caratteri e ovviamente in fase di lettura la stringa viene decodificata con la funzione SEG$ in base alla lunghezza dei campi. Questa, a dire il vero, è eccezionalmente lunga, 20 caratteri sono più che sufficienti a contenere cognome e nome, al posto degli attuali 28 e chi vorrà economizzare lo spazio in memoria potrà apportare le opportune modifiche, inoltre, cambiando i nomi dei campi, si potrà utilizzare il programma per altri scopi e trasformarlo, lasciandone immutata la struttura, in archivio per biblioteca domestica, per tenere in ordine una raccolta di dischi o per costituire uno schedario computerizzato relativo ad una raccolta di francobolli.

Qualche dettaglio in più

Il punto di partenza di un programma di questo tipo è la esatta definizione del tracciato record, cioè la definizione del numero e della lunghezza dei campi che costituiscono ogni record.
Nella fase di programmazione tale struttura non deve mai essere persa di vista e per questo va schematizzata in un modo simile a quello riportato in figura 2.
I numeri in basso indicano la posizione e, indirettamente, la lunghezza dei campi.
Per le nostre esigenze l'intero file va caricato in una matrice che nel caso specifico è stata dimensionata alla linea 220 del programma. Una matrice bidimensionale va vista come una struttura rettangolare di caselle disposte su un certo numero di righe e di colonne (vedi figura 3).
Va precisato che gli elementi della matrice, anche se non contengono nessun dato, comportano comunque un consumo di memoria. Una matrice "vuota", con cento righe e cinque colonne, occupa più di 1 Kbyte; se poi scriviamo il contenuto degli elementi dell'intera matrice consumeremo altri 12 Kbyte, arrivando al limite delle capacità del TI99/4A, con il modulo TI Extended BASIC inserito e senza l'espansione di memoria.

Il lettore si renderà conto di come sia difficile gestire in questo modo più di 100 o 200 record, pur limitando il numero e la lunghezza dei campi. Seguiamo ora la sorte di un record. Esso nasce alla linea 450 del programma; alle variabili RIG, COL e LUN vengono assegnati i valori dell'istruzione DATA di linea 410.
I contenuti dei singoli campi sono, in un primo tempo, associati alla variabile con indice CAMPO$ e solo dopo la conferma entrano nella matrice R$. Se, ad esempio, stiamo scrivendo il decimo record, sarà CTR=10 e R$(10,1) conterrà cognome e nome, R$(10,2) l'indirizzo, R$(10,3) la città, e così via. Ogni campo, in virtù della subroutine 2010, avrà una lunghezza prefissata.
Il record rimarrà nella memoria centrale e non verrà registrato su nastro se non quando sarà stato scritto l'intero file. La registrazione avviene alla riga 1730 dopo che alla linea precedente sono stati sommati gli elementi di ogni riga della matrice per formare i singoli record.
Nella fase di lettura del file, come si è già detto, avvengono le operazioni inverse: il record viene scomposto nei 5 campi per mezzo della funzione SEG$; ad esempio, SEG$(RECORD$, 57,28), restituisce il terzo campo dell'ultimo record letto, infatti si è stabilito che esso debba iniziare dal 57esimo carattere e debba esser lungo 28 caratteri. I campi così ottenuti vengono nuovamente memorizzati nella matrice R$.

La lettura è necessaria ogni qual volta si spegne il computer ma anche quando si interrompe il programma e si fa riprendere con il comando RUN perché in tal caso si annulla il contenuto della matrice R$.
Immaginiamo ora di voler accedere al nostro record in modo diretto con una ricerca basata sul nome della città (campo 3). Il nome viene accettato con l'istruzione 1350; qui LINPUT, a differenza di INPUT, accetta una stringa contenente anche eventuali virgole, senza interpretarle come separatori di variabili. Il ciclo FOR NEXT (linee 1360-1470) legge sequenzialmente tutti gli elementi della terza colonna della matrice R$ confrontandoli con il nome della città da noi specificato (variabile NOME$). Il contenuto del campo letto potrebbe essere: "20124 MILANO + space", o semplicemente: "MILANO + space" (sempre per un totale di 28 caratteri), pertanto, prima del confronto, vengono tolti gli space alla destra del nome (routine 2060). Se le due stringhe non risultassero uguali il programma le confronta nuovamente dopo aver tolto il C.A.P.
Solo quando l'esito è positivo si stampa l'intero record (vedi figura 4).
Per quanto riguarda l'Editor, si usa sostanzialmente l'istruzione CALL KEY, specificando la tastiera n. 5 con la quale vengono assegnati numeri di codice anche ai tasti di funzione. In questo modo se premiamo il tasto (D), la variabile KF specificata alla linea 2160 assume il valore 68, ma se premiamo FCTN (D) sarà KF = 9 ed in base a questo valore un'istruzione successiva farà spostare il cursore verso destra (vedi diagramma a blocchi).

Quando si preme il tasto ENTER (codice 13) il controllo passa al ciclo FOR NEXT alle linee 2240-2270, viene letta la linea editata e caricata nella variabile B$. Terminato il sottoprogramma SUB EDITOR, il contenuto di B$ passa automaticamente all'elemento corrispondente della matrice R$.
Naturalmente, se anche un solo record è stato modificato, bisognerà registrare nuovamente sul nastro l'intero file altrimenti la modifica che abbiamo apportato verrà persa al momento dello spegnimento del computer o quando il programma viene interrotto con FCTN (4) e poi rilanciato con il comando RUN.

Figura 6 - Il listato del programma.

```
100 REM
110 REM ***************************
120 REM
130 REM  RUBRICA PERSONALE
140 REM
150 REM ***************************
160 REM         E' NECESSARIO
170 REM     UN SOLO REGISTRATORE
180 REM ***************************
190 REM versione: EXT.BASIC
200 REM       1.12.1983
210 CALL CLEAR
220 DIM R$(100,5)
230 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
240 CALL CHAR(129,"7E4242424242427E"):: CALL COLOR(13,9,1)
250 DISPLAY AT(11,11):"RUBRICA" :: DISPLAY AT(13,10):"PERSONALE"
260 CALL HCHAR(9,8,128,17):: CALL VCHAR(10,8,128,5)
270 CALL VCHAR(10,24,128,5):: CALL HCHAR(15,8,128,17)
280 FOR T=1 TO 1000 :: NEXT T
290 CMP$(1)="cognome (e nome)" :: CPM$(2)="l'indirizzo"
300 CMP$(3)="la citta'" :: CMP$(4)="il num. telefonico"
310 CALL CLEAR
320 PRINT "1. SCRIVI RECORDS":"2. AGGIUNGI RECORDS"
330 PRINT "3. LEGGI RECORDS":"4. REGISTRA FILE"
340 PRINT "5. CARICA FILE":"6. FINE PROGRAMMA"
350 FOR J=1 TO 8 :: PRINT :: NEXT J
360 CALL KEY(0,K,S):: IF S=0 THEN 360
370 IF K<49 OR K>54 THEN 360 ELSE K=K-48
380 ON K GOTO 390,720,780,1640,1760,2110
390 CTR=1 :: GOSUB 1890
400 DISPLAY AT(1,17):STR$(CTR)
410 DATA 3,1,28,7,1,28,11,1,28,15,15,14,19,1,28
420 RESTORE 410
430 FOR X=1 TO 5
440 READ RIG,COL,LUN
450 ACCEPT AT(RIG,COL)SIZE(LUN):CAMPO$(X)
460 GOSUB 2010 :: NEXT X
470 DISPLAY AT(24,8):"confermi? (y/n)"
480 CALL KEY(0,K,S):: IF S=0 THEN 480
490 IF K=89 OR K=121 THEN 510
500 IF K=78 OR K=110 THEN 600 ELSE 480
510 FOR J=1 TO 5
520 R$(CTR,J)=CAMPO$(J)
530 NEXT J
540 DISPLAY AT(24,8):"fine file? (y/n)"
550 CTR=CTR+1
560 CALL KEY(0,K,S):: IF S=0 THEN 560
570 IF K=89 OR K=121 THEN 690
580 IF K<>78 AND K<>110 THEN 560
590 IF CTR=100 THEN 660
600 DISPLAY AT(3,1):" "
610 DISPLAY AT(7,1):" "
620 DISPLAY AT(11,1):" "
630 DISPLAY AT(15,15):" "
640 DISPLAY AT(19,1):" "
650 DISPLAY AT(24,1):" " :: GOTO 400
660 CALL CLEAR
670 DISPLAY AT(12,7):"* MEMORY FULL *"
680 FOR T=1 TO 1000 :: NEXT T
690 FLAGFILE=1 :: TOTREC=CTR
700 R$(CTR,1)="FINEFILE"
710 GOTO 310
720 CALL CLEAR
730 IF FLAGFILE=1 THEN 770
740 PRINT "ATTENZIONE! IL FILE NON E'":"PRESENTE IN MEMORIA.":"":"":""
750 PRINT TAB(6);"premi un tasto per":TAB(10);"continuare"
760 CALL KEY(0,K,S):: IF S=0 THEN 760 ELSE 310
770 CTR=TOTREC :: GOSUB 1890 :: GOTO 400
780 CALL CLEAR
790 IF FLAGFILE=1 THEN 830
800 PRINT "ATTENZIONE! IL FILE NON E'":"PRESENTE IN MEMORIA":"":"":"":""
```

TI 99/4A

Seguito figura 6.

```
810 PRINT TAB(6);"premi un tasto per":TAB(10);"continuare"
820 CALL KEY(0,K,S):: IF S=0 THEN 820 ELSE 310
830 DISPLAY AT(6,1):"1. LETTURA SEQUENZIALE":"":"2. LETTURA DIRETTA"
840 CALL KEY(0,K1,S):: IF S=0 THEN 840
850 IF K1=49 THEN 880
860 IF K1=50 THEN 1160 ELSE 840
870 REM *** LETTURA SEQUENZIALE ***
880 CALL CLEAR :: GOSUB 1890
890 FOR NUMREC=1 TO TOTREC-1
900 DISPLAY AT(1,17):STR$(NUMREC)
910 DISPLAY AT(3,1):R$(NUMREC,1)
920 DISPLAY AT(7,1):R$(NUMREC,2)
930 DISPLAY AT(11,1):R$(NUMREC,3)
940 DISPLAY AT(15,15):R$(NUMREC,4)
950 DISPLAY AT(19,1):R$(NUMREC,5)
960 DISPLAY AT(24,7):"modifica? (y/n)"
970 CALL KEY(0,K,S):: IF S=0 THEN 970
980 IF K=89 OR K=121 THEN 1000
990 IF K=78 OR K=110 THEN 1090 ELSE 970
1000 CALL EDITOR(R$(NUMREC,1),3,3,28)
1010 CALL EDITOR(R$(NUMREC,2),7,3,28)
1020 CALL EDITOR(R$(NUMREC,3),11,3,28)
1030 CALL EDITOR(R$(NUMREC,4),15,17,14)
1040 CALL EDITOR(R$(NUMREC,5),19,3,28)
1050 DISPLAY AT(24,7):"confermi? (y/n)"
1060 CALL KEY(0,W,S):: IF S=0 THEN 1060
1070 IF W=89 OR W=121 THEN 1090
1080 IF W=78 OR W=110 THEN 1000 ELSE 1060
1090 IF FLAG1=1 THEN 1290
1100 IF FLAG2=1 THEN 1580
1110 NEXT NUMREC
1120 CALL CLEAR
1130 DISPLAY AT(12,10):"FINE FILE"
1140 FOR TEMPO=1 TO 1000 :: NEXT TEMPO :: GOTO 310
1150 REM *** LETTURA DIRETTA ***
1160 CALL CLEAR
1170 PRINT "SCEGLI IL CAMPO CHIAVE":""
1180 PRINT "1. numero record":"2. cognome (e nome)"
1190 PRINT "3. indirizzo":"4. citta'":"5. numero di telefono"
1200 PRINT "":"":"":""
1210 CALL KEY(0,KEY,S):: IF S=0 THEN 1210
1220 IF KEY<49 OR KEY>53 THEN 1210
1230 KEY=KEY-48
1240 IF KEY>1 THEN 1340
1250 PRINT "numero record?" :: ACCEPT AT(24,1)SIZE(2)VALIDATE(DIGIT):NR
1260 IF NR>=TOTREC THEN 1510
1270 GOSUB 1890 :: NUMREC=NR
1280 FLAG1=1 :: GOTO 900
1290 FLAG1=0
1300 DISPLAY AT(24,6):"altro record? (y/n)"
1310 CALL KEY(0,K,S):: IF S=0 THEN 1310
1320 IF K=89 OR K=121 THEN 1160
1330 IF K=78 OR K=110 THEN 310 ELSE 1310
1340 PRINT "scrivi ";CMP$(KEY-1)
1350 LINPUT NOME$
1360 FOR NR=1 TO TOTREC-1
1370 GOSUB 2060
1380 IF NOME$=NOME1$ THEN 1560
1390 FOR J=1 TO LEN(NOME1$)
1400 L$=SEG$(NOME1$,J,1)
1410 IF L$=" " OR L$="/" OR L$=")" THEN 1430
1420 NEXT J :: GOTO 1470
1430 IF KEY=2 THEN 1450
1440 NOME2$=SEG$(NOME1$,J+1,LEN(NOME1$)-J):: GOTO 1460
1450 NOME2$=SEG$(NOME1$,1,J-1)
1460 IF NOME$=NOME2$ THEN 1560
1470 NEXT NR
1480 IF FLAG3=0 THEN 1510
1490 PRINT TAB(5);"OMONIMI NON PRESENTI":""
1500 FLAG3=0 :: GOTO 1520
1510 PRINT :: PRINT TAB(5);"RECORD NON PRESENTE!":""
```

```
1520 DISPLAY AT(24,5):"vuoi continuare? (y/n)"
1530 CALL KEY(0,K,S):: IF S=0 THEN 1530
1540 IF K=89 OR K=121 THEN 1160
1550 IF K=78 OR K=110 THEN 310 ELSE 1530
1560 GOSUB 1890 :: DISPLAY AT(1,17):STR$(NR)
1570 FLAG2=1 :: NUMREC=NR :: GOTO 910
1580 FLAG2=0
1590 DISPLAY AT(24,5):"cerco omonimi? (y/n)"
1600 CALL KEY(0,K,S):: IF S=0 THEN 1600
1610 IF K=89 OR K=121 THEN 1630
1620 IF K=78 OR K=110 THEN 1520 ELSE 1600
1630 FLAG3=1 :: GOTO 1470
1640 CALL CLEAR
1650 IF FLAGFILE=1 THEN 1690
1660 PRINT "ATTENZIONE! IL FILE NON E'":"IN MEMORIA.":"":"":"":""
1670 PRINT TAB(7);"<premi un tasto>"
1680 CALL KEY(0,K,S):: IF S=0 THEN 1680 ELSE 310
1690 OPEN #3:"CS1",OUTPUT,SEQUENTIAL,FIXED 128
1700 FOR NR=1 TO TOTREC
1710 RECORD$=""
1720 FOR J=1 TO 5 :: RECORD$=RECORD$&R$(NR,J):: NEXT J
1730 PRINT #3:RECORD$
1740 NEXT NR
1750 CLOSE #3 :: GOTO 310
1760 CALL CLEAR
1770 CTR=0
1780 OPEN #3:"CS1",INPUT ,SEQUENTIAL,FIXED 128
1790 CTR=CTR+1
1800 LINPUT #3:RECORD$
1810 R$(CTR,1)=SEG$(RECORD$,1,28)
1820 R$(CTR,2)=SEG$(RECORD$,29,28)
1830 R$(CTR,3)=SEG$(RECORD$,57,28)
1840 R$(CTR,4)=SEG$(RECORD$,85,14)
1850 R$(CTR,5)=SEG$(RECORD$,99,28)
1860 IF SEG$(R$(CTR,1),1,8)<>"FINEFILE" THEN 1790
1870 FLAGFILE=1 :: TOTREC=CTR
1880 CLOSE #3 :: GOTO 310
1890 CALL CLEAR :: DISPLAY AT(1,10):"record: "
1900 DISPLAY AT(4,1):RPT$("_",28)
1910 DISPLAY AT(5,1):"cognome e nome"
1920 DISPLAY AT(8,1):RPT$("_",28)
1930 DISPLAY AT(9,1):"indirizzo"
1940 DISPLAY AT(12,1):RPT$("_",28)
1950 DISPLAY AT(13,1):"(cap) citta'"
1960 DISPLAY AT(16,15):RPT$("_",14)
1970 DISPLAY AT(17,15):"tel."
1980 DISPLAY AT(20,1):RPT$("_",28)
1990 DISPLAY AT(21,1):"note"
2000 RETURN
2010 IF LEN(CAMPO$(X))=LUN THEN 2050
2020 FOR J=1 TO LUN-LEN(CAMPO$(X))
2030 CAMPO$(X)=CAMPO$(X)&" "
2040 NEXT J
2050 RETURN
2060 FOR J=LEN(R$(NR,KEY-1))TO 1 STEP -1
2070 IF SEG$(R$(NR,KEY-1),J,1)<>" " THEN 2090
2080 NEXT J
2090 NOME1$=SEG$(R$(NR,KEY-1),1,J)
2100 RETURN
2110 CALL CLEAR :: END
2120 SUB EDITOR(B$,R,I,L)
2130 B$=""
2140 X=I*8-7 :: Y=R*8-7 :: PP=X :: UP=X+(L-1)*8
2150 CALL SPRITE(#1,129,7,Y,X)
2160 CALL KEY(5,KF,ST):: IF ST=0 THEN 2160
2170 IF KF<>8 AND KF<>9 AND KF<>13 THEN 2220
2180 IF KF=8 THEN IF X>PP THEN X=X-8
2190 IF KF=9 THEN IF X<UP THEN X=X+8
2200 IF KF=13 THEN 2240
2210 CALL LOCATE(#1,Y,X):: GOTO 2160
2220 COLO=INT(X/8)+1 :: CALL HCHAR(R,COLO,KF)
2230 IF X<UP THEN X=X+8 :: GOTO 2210
2240 FOR J=I TO I+L-1
2250 CALL GCHAR(R,J,CARAT)
2260 B$=B$&CHR$(CARAT)
2270 NEXT J
2280 CALL DELSPRITE(#1)
2290 SUBEND
```

Seguito figura 6.

Prova riflessi e Simon

di **Sandro Del Bello**
e **Anna Maria Paganini**

Il primo programma consiste di tre prove, al termine delle quali il giocatore potrà avere una valutazione dei suoi riflessi.
Prima prova: si dovrà, nel minor tempo possibile, schiacciare il tasto corrispondente alla lettera che compare al centro del display.
Questa prova risulta piuttosto impegnativa in quanto non è facile riuscire a compiere l'operazione nel tempo massimo consentito; per questo si consiglia di partire con le mani già vicine alla tastiera.
Meno velocemente verrà premuto il tasto, maggiori saranno le penalità assegnate, fino ad un massimo di dieci.
Seconda prova: sul display si susseguono rapidamente le cifre dallo zero al nove; la prova riflessi consiste nel premere con decisione il tasto del nove nello stesso istante in cui tale numero compare.
Ciò andrà eseguito per tre volte a differenti velocità. Ad ogni errore del giocatore verranno sommate al punteggio complessivo cinque penalità.
Terza prova: un quadratino salterà da uno spazio all'altro del display ed il giocatore dovrà fermarne la corsa premendo il numero, dallo zero al nove, corrispondente allo spazio in cui momentaneamente si trova. Qui non vi è un massimo di penalità, in quanto la prova non termina fino a quando il giocatore non avrà bloccato il quadrato; però ogni salto costerà cinque punti di penalità.
Al termine delle tre prove comparirà il punteggio totale conseguito e la valutazione associata, da pessimo a ottimo.

Il secondo programma è una variante del gioco del Simon. Questa versione consiste nel ricordare e ripetere un numero sempre maggiore di cifre, fino all'inevitabile errore. All'inizio il computer chiederà il nome del giocatore per verificare la sua eventuale presenza nella classifica dei cinque migliori punteggi fino a quel momento contenuti in memoria.
In caso positivo verrà visualizzato il record personale che il giocatore ha ottenuto precedentemente. Al termine della prova, se avrete ottenuto un punteggio superiore a quello minimo attuale, la classifica verrà aggiornata con un eventuale vostro inserimento. Dopo un po' di pratica non è difficile arrivare intorno alle venti, venticinque cifre memorizzate; i bravissimi manderanno in tilt il calcolatore se supereranno le trenta.

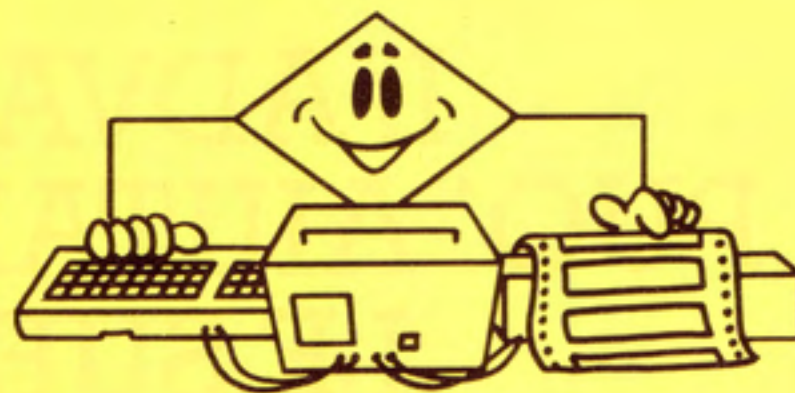
Commenti al listato n° 1

Il listato occupa tutta la memoria disponibile. Per questo le valutazioni dei riflessi andranno inserite in modo RUN.
La "subroutine 30" che si incontra frequentemente ha la funzione di pausa variata volta per volta tramite il parametro L.
Il programma dovrebbe risultare sufficientemente leggibile. Si può chiarire che $ dapprima contiene le ventuno lettere dell'alfabeto italiano, fra le quali verrà casualmente selezionata quella che comparirà sul display; in seguito, e per tutto il programma, tale variabile conterrà la stringa di carattere "penalità".
P, Q e D indicano le penalità totalizzate rispettivamente nella prima, seconda e terza prova, mentre W rappresenta la loro somma.
Riga 7: il ciclo su F seleziona le tre differenti velocità della prova.
Riga 16: l'uso dei due IF consecutivi consente di interrompere il ciclo soltanto nel caso in cui il bersaglio è stato colpito.
Righe 19-20: le penalità complessive vengono divise per otto per assegnare la valutazione del punteggio ottenuto.

Commenti al listato n° 2

Righe 10-40: per i soliti motivi di spazio vengono considerate solo le prime sette lettere del nome del giocatore.
Da notare che F = 1 se il giocatore non è già in classifica.
Righe 50-210: scelta casuale della cifra da aggiungere alla sequenza. Il modo macchinoso con cui è fatta l'operazione dipende dal fatto che $ è già occupato dalle cifre precedenti.
Righe 220-250: calcolo del punteggio minimo per verificare l'entrata in classifica.
Righe 250-270: in caso positivo, se il giocatore era già presente, viene aggiornato il suo record.
Righe 275-295: stampa della classifica e ... via per un'altra prova.

Listato 1 - Il programma del Prova riflessi.

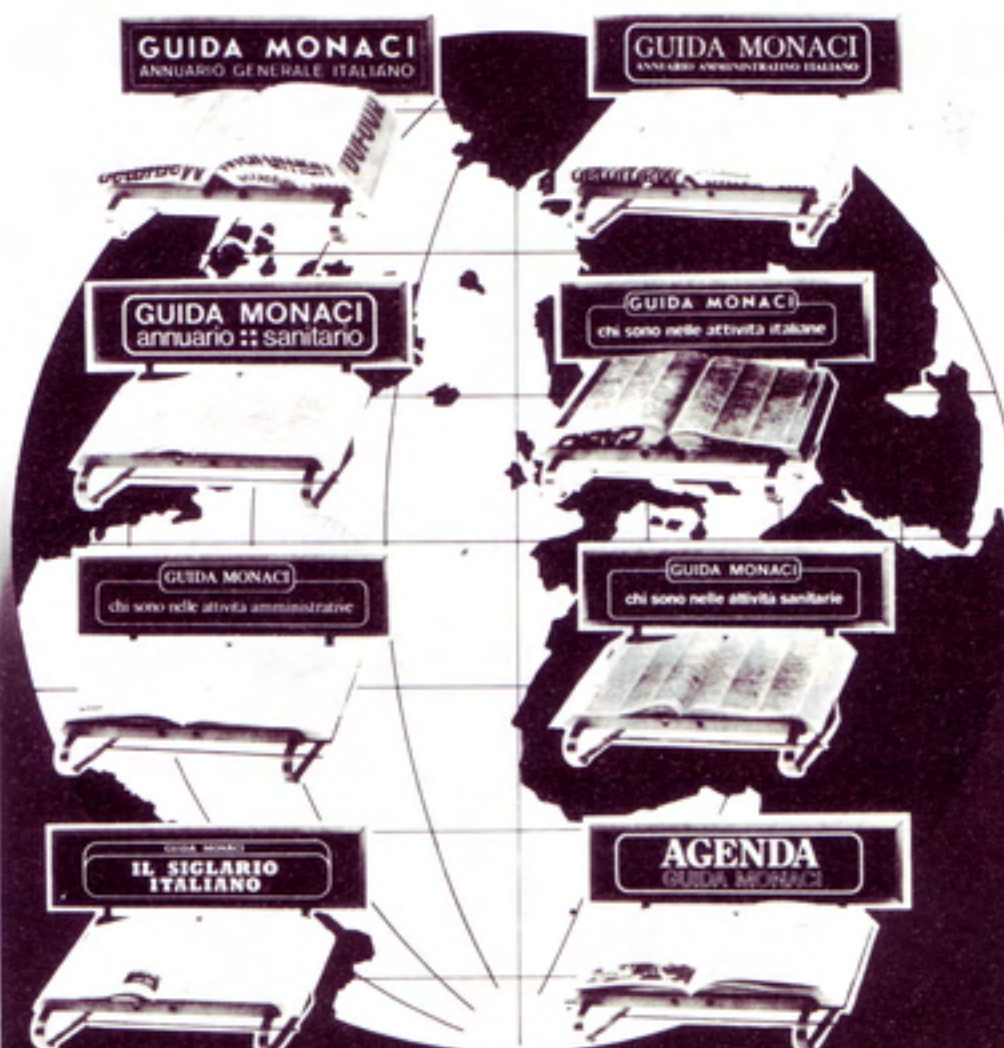
```
FUORI PROGRAMMA IN MODO RUN

R$(0) = "OTTIMO"
R$(1) = "BUONO"
R$(2) = "NORMALE"
R$(3) = "SCARSO"
R$(4) = "PESSIMO"


1  PRINT "RIFLESSI"::L=200: GOSUB 30
2  Q=0 : D=0 ; $ =
"ABCDEFGHILMNOPQRSTUVZ" : H=INT(RAN # *
21) +1
3  A$=MID(H,1) : $ = "Penalita'" :N$ =
"TOTALE=": PRINT "Prova no.1"
4  GOSUB 30 : PRINT CSR6;A$;
5  FOR O = 1 TO 39 : IF KEY ≠ A$:NEXT
```

Seguito listato 1.

```
0:PRINT CSR0;"Troppo tardi";
6  P=INT(O/4)
7  PRINT P;$;: GOSUB 30: PRINT "Prova
no.2": FOR F=15 TO 5 STEP -5
9  FOR C=0 TO 9:L=F:GOSUB 30: PRINT
CSR6;C;
10 IF KEY ≠ "9"; NEXT C : GOTO 9
11 L=150:IF C = 9; GOSUB 30: PRINT CSR0;
"Bene! 0";$;: GOTO 13
12 GOSUB 30: PRINT CSR0; "No! 5";$;:
Q=Q+5
13 L = 200: GOSUB 30 : NEXT F: PRINT N$;
Q;: GOSUB 30
14 PRINT "Prova no.3": GOSUB 30
15 A = INT(RAN #*10): PRINT CSRA;"■";
16 FOR I = 1 TO 25: B$ = KEY:IF B$ ≠ "";
IF VAL(B$) = A THEN 18
17 NEXT I:PRINT:D=D+5:GOTO 15
18 D = D +INT (I/5):W = P + Q + D : PRINT
CSR0; "  Colpito! ";D;$,N$;W;
19 GOSUB 30: IF W > 30; W = 39
20 PRINT R$(INT(W/8)): GOTO 2
30 PRINT:FOR O=1 TO L:NEXT O:RETURN
```

Listato 2 - Il listato del gioco del Simon.

```
10 F=0:INPUT"NOME",$
15 IF LEN($)<7;N$=$:GOTO 25
20 N$=MID(1,7)
25 FOR I=1 TO 9 STEP 2
30 IF N$=N$(I);PRINT "IL TUO RECORD
```

Seguito listato 2.

```
E'";N(I+1);:GOTO 50
40 NEXT I:F=1
50 PRINT " ■ PREMI EXE":$=""
70 A=INT(5*RAN#)
80 IF A=0; D$="1"
90 IF A=1;D$="2"
100 IF A=2;D$="3"
110 IF A=3;D$="4"
120 IF A=4;D$="5"
130 $=$+D$:L=LEN($)
140 FOR K=1 TO L:J$=MID(K,1)+"    "
150 PRINT J$;:NEXT K
160 PRINT "RIPETI";
170 FOR K=1 TO L
180 INPUT Z$
190 IF Z$ ≠ MID(K,1);PRINT "ERRORE ALLA
MOSSA";L:GOTO 220
210 NEXT K:PRINT "PROSEGUIAMO     ";:GOTO
70
220 M=N(2):H=2
230 FOR K=4 TO 10 STEP 2
240 IF N(K)<M;M=N(K):H=K
250 NEXT K:IF F ≠ 1 THEN 270
260 IF L>M+1;N$(H-1)=N$:N(H)=L-1
265 GOTO 275
270 IF L>N(I+1)+1;N(I+1)=L-1
275 PRINT "Classifica  ";
280 FOR C=1 TO 9 STEP 2
290 IF N$(C)≠"";PRINT N$(C);N(C+1)
295 NEXT C:GOTO 10
```

L'ADVANCE 86A /START PUO' AIUTARLA A RADDOPPIARE LE SUE VENDITE DI PC

PERCHÈ È L'UNICO VERO PC 16 BIT A SOLO 1.400.000*

CON MICROPROCESSORE 8086 E CON
- **128 O 256KB DI MEMORIA UTENTE CON CONTROLLO DI PARITA'**
- **40KB DI MEMORIA ROM**
- **MEMORIA GRAFICA A 16 COLORI**
- **ALTOPARLANTE INCORPORATO**
- **HARD E SOFTWARE 100% IBM/PC COMPATIBILE**
- *DIAGNOSTICA, BASIC E SISTEMA OPERATIVO PER CASSETTA CONTENUTO NELLA ROM*
  - *IL BASIC COMPRESO*
  - *UN SET DI 256 CARATTERI IN ROM*
  - *VISUALIZZAZIONE TV, RGB E MONITOR COMP/SYNC A COLORI O MONOCROMATICO*
  - *COMPLETA GESTIONE DEL VIDEO*
  - *QUATTRO PAGINE DI VIDEO*
- *TESTO 80x25 O 40x25*
- *RISOLUZIONE GRAFICA 300x200 O 640x200*
- *PORTA PER CASSETTA, PER PENNA OTTICA, PER PENNA GIOCHI (JOYSTICK) E CON INTERFACCIA CENTRONICS*

IN PIU' L'ADVANCE 86A /START PUO' UTILIZZARE STAMPANTI DI QUALUNQUE TIPO PURCHE' UTILIZZINO INTERFACCIA CENTRONICS; E' ESPANDIBILE DALL'UTENTE NEL MODELLO CON 2 FLOPPY DA 320KB CIASCUNO O NEL MODELLO CON HARD DISK DA 10MB; PUO' ESSERE CORREDATO DI COPROCESSORE ARITMETICO 8087; UTILIZZA CUSTOM CHIPS ED ELETTRONICA DELLA FERRANTI INSTRUMENTS E DRIVES DELLA SHUGART ASSOCIATES.

QUANDO DI UN PC SI PUO' DIRE TUTTO QUESTO NON E' NECESSARIO AGGIUNGERE ALTRO.

CONDOR INFORMATICS ITALIA SRL
VIA GRANCINI 8
20145 MILANO
TEL. 02/43.45.62-49.87.549-49.87.713

Chiunque desideri avere informazioni su un'eventuale concessione di vendita può telefonare o restituire questo tagliando.

BT

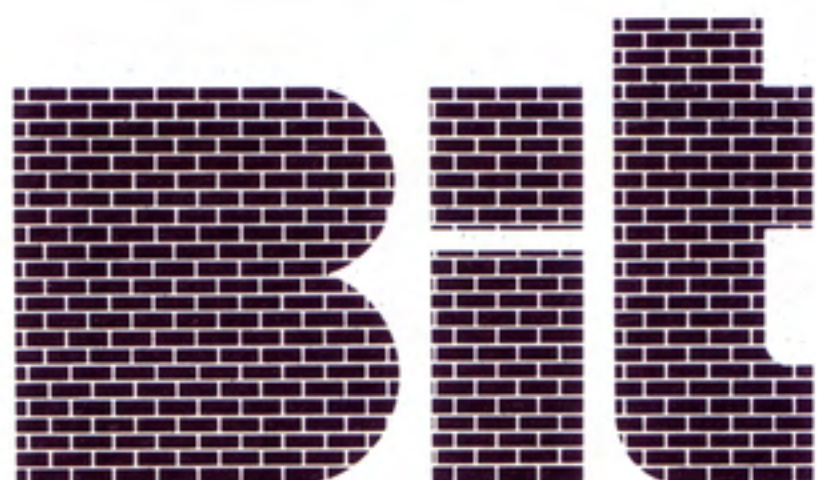
NOME ____________________

SOCIETA' ____________________

INDIRIZZO ____________________

CITTA' ____________ TEL. ____________

*IN FUNZIONE DEL CAMBIO DEL DOLLARO

SERVIZIO SOFTWARE

Bit propone ai propri lettori
i dischi o le cassette
dei programmi pubblicati.
I programmi,
provati e garantiti,
sono di immediato utilizzo.

P.S. n°	Programma	Sistema	Prezzo	Codice	Supporto
38	Gioco della scimmia Spaccamattoni	VIC 20	15.000	VI381A	Cassetta
38	Planel	Apple II	20.000	AP382C	Disco
39	Rompicapo di Rubik	CBM 4032 CBM 3032	15.000 20.000	PE391A PE392B	Cassetta Disco
39	Breakout	CBM 3032	20.000	PE393A	Cassetta
42	Apple-Chef	Apple II	20.000	AP422C	Disco
42	Provariflessi	VIC 20	15.000	VI421A	Cassetta
45	Tiny FORTH	Apple II	35.000	AP452A	Disco
45	Alì Babà	ZX Spectrum	15.000	SP451B	Cassetta
45	1X2	PET 3032	15.000	PE451C	Cassetta

Per richiedere i programmi in contrassegno, pagando direttamente al postino la cifra indicata, inviare il seguente tagliando

Spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Inviatemi i seguenti nastri e/o dischi con i programmi pubblicati su Bit.

Cod. ☐☐☐☐☐☐ **a L.** ....

Cod. ☐☐☐☐☐☐ **a L.** ....

Cod. ☐☐☐☐☐☐ **a L.** ....

Cod. ☐☐☐☐☐☐ **a L.** ....

Spese postali (contributo fisso) L. 2.000

TOTALE L. ....

che pagherò al postino alla consegna del pacco.

GRUPPO EDITORIALE JACKSON

Cognome ....

Nome ....

Indirizzo ....

CAP ....

Città ....

Firma ....

Il Questionario di Bit

Caro Lettore,
ti chiediamo lo sforzo di riempire questo modulo con dati esatti e di spedircelo al più presto. L'obiettivo è quello di conoscere meglio la tua figura e le tue esigenze allo scopo di migliorare la nostra rivista e fornirti un servizio più accurato e puntuale. Perciò non essere affatto diplomatico nelle tue critiche: ti raccomandiamo solo di non riferirti ai pur seri difetti episodici o a quelli di natura organizzativa, in quanto sono già a nostra conoscenza e ci vedono da tempo impegnati ad ovviarli. Insomma pensa prevalentemente alle questioni di tipo tecnico e concettuale, all'impostazione culturale, al grado di approfondimento desiderato e così via.
La tua tipologia è pure della massima importanza, in quanto se accontentare tutti è impresa ardua per chiunque, cercar di seguire i desideri della maggioranza costituisce pur sempre il migliore dei modi possibili nel più democratico dei mondi possibili.
A quanti hanno a cuore la riservatezza concediamo di omettere le proprie generalità personali, mentre tutti i "coraggiosi" che, per di più, riusciranno entro un mese a risolvere i tre piccoli microquiz proposti, possono contare nella benevolenza dell'Editore, sotto forma di buono-sconto per l'acquisto di pubblicazioni del Gruppo Editoriale Jackson.

La Redazione

SEZIONE A — GENERALITA'

Cognome ....
Nome ....
Via .... n. ....
Città .... CAP. .... Prov. .... Tel. ....

1. Età
- ☐ sotto i 15 anni
- ☐ tra i 15 e i 20
- ☐ tra i 20 e i 30
- ☐ tra i 30 e i 45
- ☐ oltre i 45 anni

(barrare la casella che serve)

2. Professione ....

2.1. Il settore in cui operi ha attinenza con l'Informatica?
- ☐ SI
- ☐ NO

In caso affermativo precisane il ramo d'attività:
vendita o assistenza tecnica di grandi elaboratori, di mini, di microcomputer; centro elaborazione dati; altro (specificare)

....

2.2. Possiedi un personal computer (anche in società con amici)?
- ☐ SI
- ☐ NO

Se SI indicane la marca ed il modello:

....

2.3. Usi un elaboratore sul posto di lavoro?
- ☐ SI
- ☐ NO

Se SI, si tratta di un personal?
- ☐ SI
- ☐ NO

SEZIONE B — BACKGROUND

1. Come definiresti la tua preparazione in Informatica? (scegli tra scarsa, mediocre, sufficiente, buona)

....

2. Qualunque ne sia il grado, tale preparazione da cosa deriva?
- ☐ da studi fatti presso una scuola secondaria (ITI, ITC)
- ☐ dalla frequenza di un Corso di laurea in Scienza dell'Informazione
- ☐ idem c.s. per Ingegneria elettronica
- ☐ da un corso universitario
- ☐ da corsi aziendali
- ☐ da corsi privati
- ☐ da studi personali.

3. Quali linguaggi di programmazione conosci in modo non troppo superficiale? (indicarne anche più d'uno)
- ☐ BASIC
- ☐ Pascal
- ☐ FORTRAN
- ☐ COBOL
- ☐ FORTH
- ☐ Logo
- ☐ altro (specificare):

....

4. Quali di questi "package" conosci ed utilizzi, almeno saltuariamente?
- ☐ un word processor (specificare)

  ....
- ☐ VisiCalc
- ☐ Multiplan
- ☐ Personal Data Base
- ☐ dBase II
- ☐ altro (specificare):

....

5. Possiedi qualche conoscenza di elettronica e di hardware?
- ☐ SI
- ☐ NO (o molto vaga)

SEZIONE C — INTERESSI

1. Come definiresti il tuo interesse per il personal computer? (segnare eventualmente anche due caselle)
- ☐ amatoriale
- ☐ di studio
- ☐ professionale
- ☐ altro (specificare):

....

2. Come utilizzi prevalentemente il personal computer? (segna una sola casella)
- ☐ per diletto
- ☐ per lavoro

3. Dove adoperi normalmente il personal computer?
- ☐ a casa
- ☐ in ufficio
- ☐ a scuola
- ☐ altro (specificare):

....

Il Questionario di Bit

4. Sei iscritto a qualche club di personal computer?
- ☐ SI
- ☐ NO

Se SI specificare quale:

....

5. A quale di questi argomenti sei maggiormente interessato (segnarne al massimo 3)
- ☐ hardware
- ☐ software di base e linguaggi
- ☐ programmi pratici
- ☐ principi d'Informatica e criteri teorico-pratici di programmazione
- ☐ recensioni di sistemi e periferiche
- ☐ giochi
- ☐ package e loro uso
- ☐ altro (specificare):

....

SEZIONE D — GIUDIZIO SU BIT

1. Dovendo dare un giudizio globale sulla comprensibilità della rivista, come ti esprimeresti?
- ☐ è scarsamente professionale
- ☐ ha a volte argomenti difficili ma spiegati per lo più chiaramente
- ☐ è troppo scarso lo spazio per principianti
- ☐ è fatta per soli iniziati
- ☐ altro (specificare):

....

2. Recentemente, lo avrai notato, nella "Vetrina" si è introdotto un certo spazio per notizie su varie manifestazioni legate all'informatica, anche al di fuori delle fiere e mercati (esempio, Mostra "Jarry e la Patafisica", Convegno su Microinformatica e medicina, interviste ad operatori del settore ecc.). Cosa ne pensi (dai un breve giudizio analitico)?

....

....

....

3. Una novità '83 è stata poi l'istituzione del fascicolo Riservato Personal.
A parte la qualità (su cui voterai più avanti) che giudizio esprimi su tale iniziativa?
- ☐ andrebbe ampliata
- ☐ valida senza condizioni
- ☐ ruba troppo spazio a cose più interessanti
- ☐ non mi interessa affatto

4. Esprimi un voto da 1 a 10 sulla qualità delle varie sezioni o rubriche:

EDITORIALE	....
BIT FLASH	....
VETRINA	....
HARDWARE	....
BITEST	....
SOFTEST (package)	....
SOFTWARE (linguaggi e criteri)	....
SOFTWARE (applicazioni)	....
RICETTARIO	....
FEEDBACK	....
SUPERBIT/RIS. PERSONAL	....
DIGIDATTICA	

5. Se dipendesse da te, quale rubrica aboliresti?

....

6. Cita due rubriche che, invece, vorresti ampliate e potenziate:

a) .... b) ....

7. Precisa infine due argomenti che ti piacerebbe venissero più di sovente trattati da **Bit**.

....

SEZIONE E — OSSERVAZIONI E SUGGERIMENTI

1. Utilizza infine queste poche righe per segnalarci liberamente, desideri, miglioramenti, idee.

....

....

....

....

2. Se ritieni di poter collaborare alla rivista, indica qui sotto il tuo campo di competenza (per una proposta più dettagliata scrivici):

....

....

Desideri l'invio della nostra Guida agli autori? (si/no) ....

I TRE QUIZZULLI, SE TI TRASTULLI

QUIZZULLO n. 1 - Scrivi l'iniziale del nome del fondatore, a soli 18 anni, della Microsoft (cognome: Gates);
QUIZZULLO n. 2 - Crivello di Eratostene, Indovinello della Sfinge, Bubble sort: quale di queste tre cose non merita il nome di algoritmo (indicare la sola lettera iniziale);
QUIZZULLO n. 3 - Tra questi linguaggi (FORTRAN, BASIC, FORTH) quale supporta la ricorsività? Scrivere stavolta la penultima lettera.
In definitiva, unendo di seguito le tre lettere trovate, ottengo la sigla:

....

Vivissimi ringraziamenti a quanti son giunti fin qui.

Data ....

Da staccare e inviare a:
Redazione di Bit "Il questionario"
Gruppo Editoriale Jackson
Via Rosellini, 12
20124 Milano

Agendina

di **Franco Antichi**

Il programma è per il piccolo Casio PB 100 che è utilizzabile sia con la versione normale che, meglio, con la versione avente la memoria espansa. Si tratta di un notes o agenda che può trovare anche altre applicazioni, specie con le aggiunte che indichiamo in coda al listato.
Vengono utilizzate le memorie che vanno dalla Z(1) in poi in modo che il PB 100 resta disponibile e utilizzabile per altri programmi, senza interferenze o sovrapposizioni di sorta. Nel mio caso ho utilizzato un'espansione di 24 memorie (fare DEFM 24) che possono essere più o meno estese a seconda della necessità (basta digitare DEFM n e variare le linee 30 e 60). Una volta iniziato il programma, è sufficiente premere EXE e compaiono in successione i vari messaggi registrati. Per inserirne di nuovi, dopo premuto EXE, nell'intervallo che intercorre prima dell'apparizione di altri messaggi, si deve premere "I" (inserimento) e comparirà il "?" della richiesta di INPUT; si scrive quindi il messaggio (massimo 30 lettere). Analogamente per cancellare la scritta che appare sul display è sufficiente premere EXE e poi subito dopo "C" (cancella). Il programma può essere variato per far uscire i messaggi in successione continua (linea 20 aggiungere un ;) ma a mio giudizio con meno funzionalità. Al termine dei messaggi appare la scritta, che può essere omessa, "me n°", che indica il numero di memorie ancora disponibili. Quando le memorie di output raggiungono le 24, compare la scritta "FULL". Inserendo un GOTO 7 alla linea 30 (togliere END) il ciclo può continuare indefinitamente. Aggiungendo le linee citate in coda è possibile l'output dei soli messaggi che contengono i dati richiesti. (esempio, con input A escono i messaggi contenenti una A, ecc.).

Non ho contato i passi del programma; ritengo però siano meno di 400. Perciò con il PB 100 espanso a 1568 passi, si hanno circa 1200 passi restanti pari a 150 memorie di 7 lettere = 1050 lettere in totale; rimane quindi ancora molto spazio per chi vuole utilizzare il PB 100 negli altri nove programmi.

Commenti

Input - Le linee 50-80 servono per l'input dei messaggi. Il messaggio viene caricato in $ e poi vengono contate le lettere; se il numero delle lettere è multiplo di 7 viene inserito uno spazio in coda, che servirà da segnale di fine messaggio. Vengono contate le variabili da usare (x) e caricate nella $ con il ciclo FOR-NEXT delle linee 65-80.
Output - Le linee 7-30 servono per l'output: le variabili vengono sommate in $ fino a quando (linea 10 - IF ecc.) viene aggiunta la variabile con meno di 7 lettere; a questo punto il messaggio viene stampato. Il ciclo 21-26 serve per dar tempo all'operatore di decidere se inserire, cancellare o meno e può essere allungato o accorciato variando la linea 21.
Cancellazione - Le linee 100-125 servono per la cancellazione: viene cancellato il messaggio che compare in quel momento sul display. Il "buco" delle variabili cancellate viene riempito spostando verso la Z(1) tutte le restanti variabili Z$. Per inizializzare il programma occorre fare Z(1)=1 e RUN 50 inserendo la prima parola (esempio AGENDA) poi si procede come detto senza altri accorgimenti.
Estensione del programma - Con l'aggiunta delle linee in coda al listato si può ottenere l'uso forse più interessante di questo programma con l'uscita di dati particolari. Per esempio, inputando un nome, anche parzialmente, od una lettera oppure un numero, il PB 100 analizza tutti i messaggi presenti e stampa solo quelli contenenti tale lettera o dato. Può essere utilizzato così come agenda telefonica i cui nomi possono essere scritti a caso e fatti comparire in ordine alfabetico, o come un elenco prezzi (si scrive il nome e compare nome e prezzo), o come piccolo dizionario di termini tecnici.

```
7   S = " "
10  FOR T = 2 TO Z(1) : $ = $ + Z$ (T) :
IF FRAC (LEN ($)/7) = Ø THEN 2Ø
12  GOTO 3Ø
2Ø  PRINT $
21  FOR A = 1 TO 25
23  IF KEY = "C" THEN 1ØØ
(cancellazione)
24  IF KEY = "I" THEN 5Ø
(inserzione)
26  NEXT A
27  $ = " "
3Ø  NEXT T : PRINT "me" ; 24-Z(1); : END

5Ø  INPUT $ : GOSUB 35Ø : V = -6: U = 7
6Ø  IF Z(1) + X + 1 ≥ 24; PRINT "FULL" :
END
65  FOR T = 1 TO X: V = V + 7 : IF T = X:
U = Y - (X - 1) *7
7Ø  Z(1) = Z(1) + 1 : Z$(Z(1)) = MID
(V,U)
8Ø  NEXT T : Z$(Z(1)+1) = "" : GOTO 7

1ØØ GOSUB 35Ø
11Ø FOR A = 1 TO X
12Ø FOR B = T - X + 1 TO Z(1) : Z$(B) =
Z$(B+1): NEXT B
125 Z(1) = Z(1) -1: NEXT A : GOTO 7

35Ø Y = LEN($): IF FRAC Y/7 = Ø; $ = $ +
" ": Y = LEN($)
351 X = INT(Y/7 + Ø,99) : RETURN
```

Listato 1 - Per iniziare battere: Z(1)=1 DEFM 24 (o più). Il primo messaggio si scrive usando RUN 50.

```
5   E$ = ""
25  IF KEY = "U" THEN 19Ø
1Ø  ............. THEN 13
3Ø  ............. : E$ = ""
8Ø  ............. GOTO 5
13  IF E$ = "" THEN 2ØØ

19Ø INPUT E$
2ØØ Y = LEN ($) : Z = LEN (E$)
21Ø FOR A = 1 TO Y: IF Z = 1; IF Y-A+1 <
Z; $ = "": GOTO 3Ø
22Ø IF MID(A,Z) = E$ THEN 2Ø
23Ø NEXT A:$ = "": GOTO 3Ø
```

Listato 2 - Programma per output selezionato. Inserendo un dato escono tutti i messaggi che lo contengono. Variare il listato 1 come riportato in queste linee.

Modelli simulatori

Una fantastoria che vi guida passo passo all'uso di un programma che permette di realizzare il grafo di un progetto e poi procedere alla simulazione, utilizzando HP 86A e la stampante 82905B.

di **Francesco Moscarella**

Antefatto

Forger cercò di dare ordine al turbine di pensieri che lo assalivano. Man Friday era stato più prezioso di quello che aveva previsto ed aveva portato via con sé buona parte della competenza e capacità decisionale del suo Consiglio di Amministrazione.
Solo in quel momento la realtà gli risultò evidente riflessa nei volti inespressivi che coronavano il lungo tavolo di ebano.
Pensò che forse negli ultimi anni aveva fatto pesare troppo la convenienza politica nella scelta dei suoi collaboratori. Questa volta avrebbe dovuto decidere da solo.
Sciolse il Consiglio e si diresse verso il suo ufficio, attraversando corridoi che, data l'ora, erano vuoti e silenziosi.

Seduto nella sua poltrona, fece in modo che la preoccupazione fosse sostituita dalla determinazione e fiducia nei propri mezzi che erano le leve principali del suo successo nella vita.

Le definizione del problema

Con pochi tratti schematizzò la nuova linea di produzione, rappresentando con quadrati gli otto reparti che la formavano. Tracciò poi cinque cammini fra i reparti ad indicare i differenti flussi di lavorazione previsti (vedi figura 1).
Stimò la capacità produttiva, i costi ed i volumi. Cercò poi di prevedere gli effetti di un lungo periodo produttivo sulle code e sulle efficienze dei reparti.
Gli parve di avere individuato alcuni punti critici; ma anche dopo lunga riflessione, non riuscì ad essere sicuro di aver raggiunto un risultato concreto.
Si ricordò allora di un noiosissimo studio che aveva ricevuto dal gruppo dei modelli matematici, quei "saputissimi" individui che nella pratica non avrebbero saputo piantare un chiodo in un muro.
Si ricordava vagamente di alcuni grafici simili a quelli che aveva appena disegnati. Trovò, non senza fatica, il plico in questione quando ormai disperava di riuscirvi.
Rilesse il rapporto con cura e si convinse di avere nelle mani qualcosa di utile.
Era solo in azienda ma fidava di sapersela cavare da solo. Prese il disco allegato al plico, ruotò la sua poltrona di 90 gradi e attivò il suo HP 86A.

Procedura di utilizzo del programma

La key #1
Al video comparve il menu con le sue chiavi (vedi figura 2).
Forger premette la key #1 per accingersi a definire il modello. Il programma richiese il nome del progetto e i parametri produttivi di ciascun reparto (vedi figure 3 e 4).

Figura 1 - Esempio grafico di un impianto formato da 8 reparti produttivi. Ciascuna attività (reparto) è individuata dalle sue coordinate nella matrice di 17★7 caselle, e può prevedere fino a due uscite.

```
key#1:preparazione modello simulatore
key#2:rappresentazione grafica del modello

key#3:simulazione

key#4:risultati statistici della simulazione
key#5:descrizione attivita'

key#7:fine del programma
```

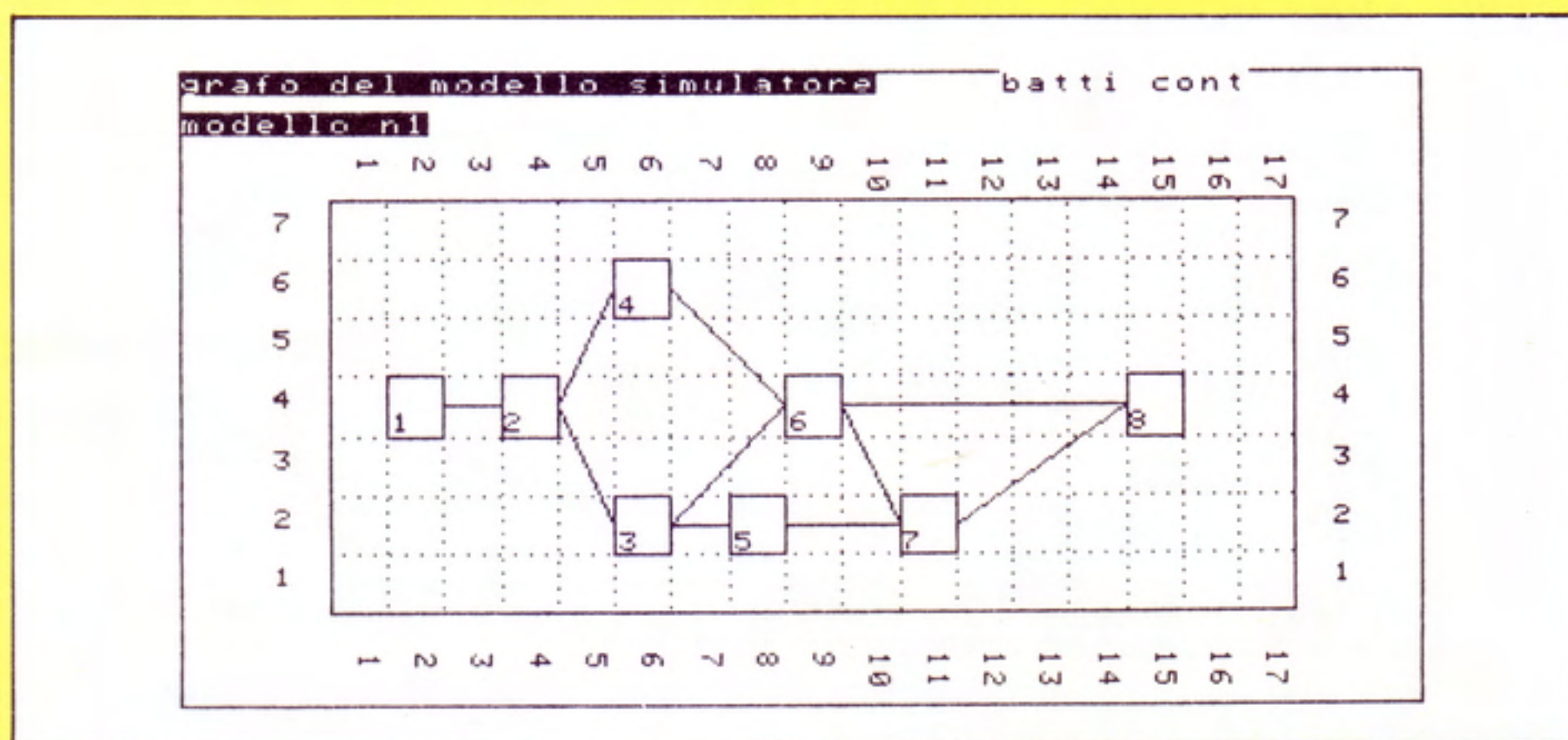



Figura 2 - Le sei key del programma con le relative opzioni.

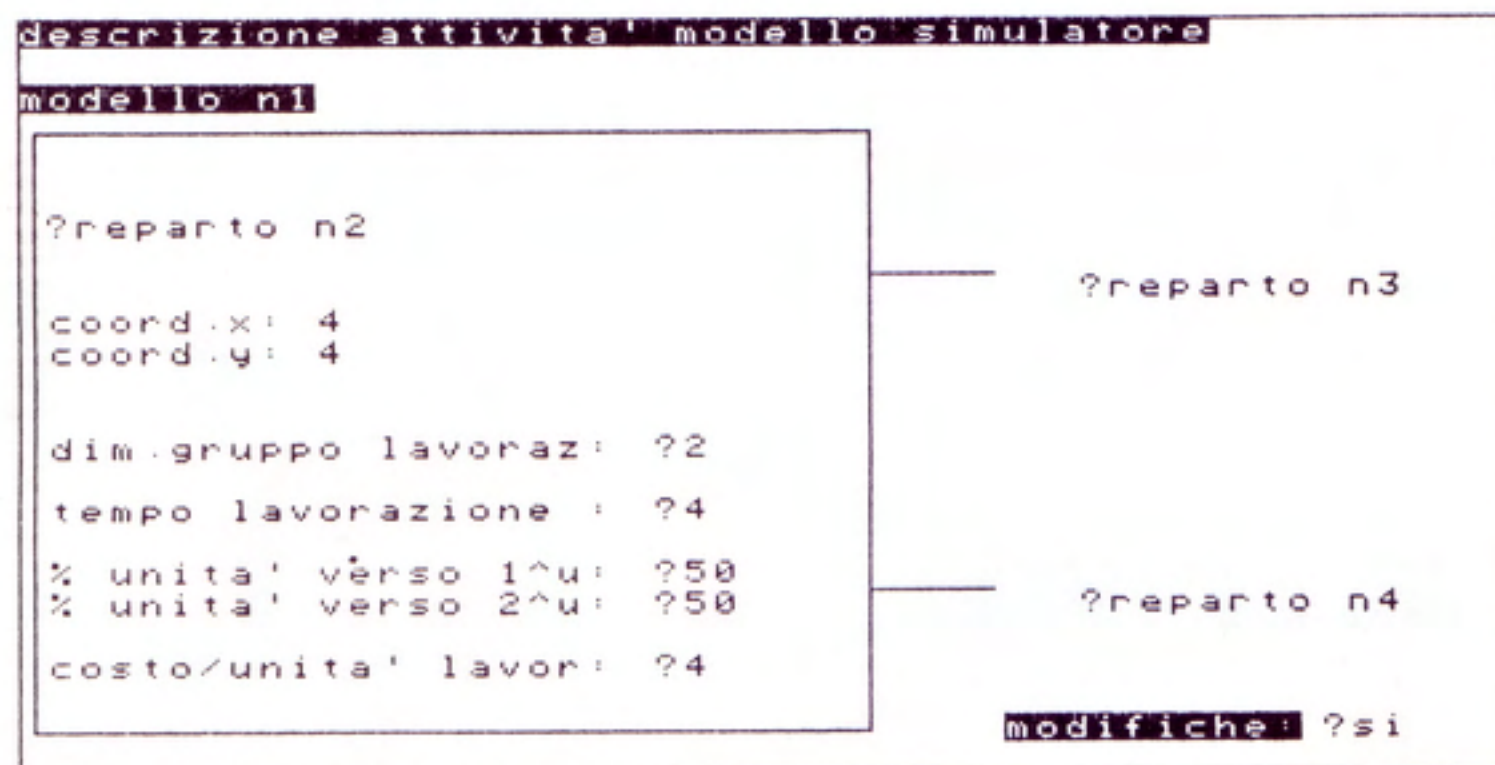
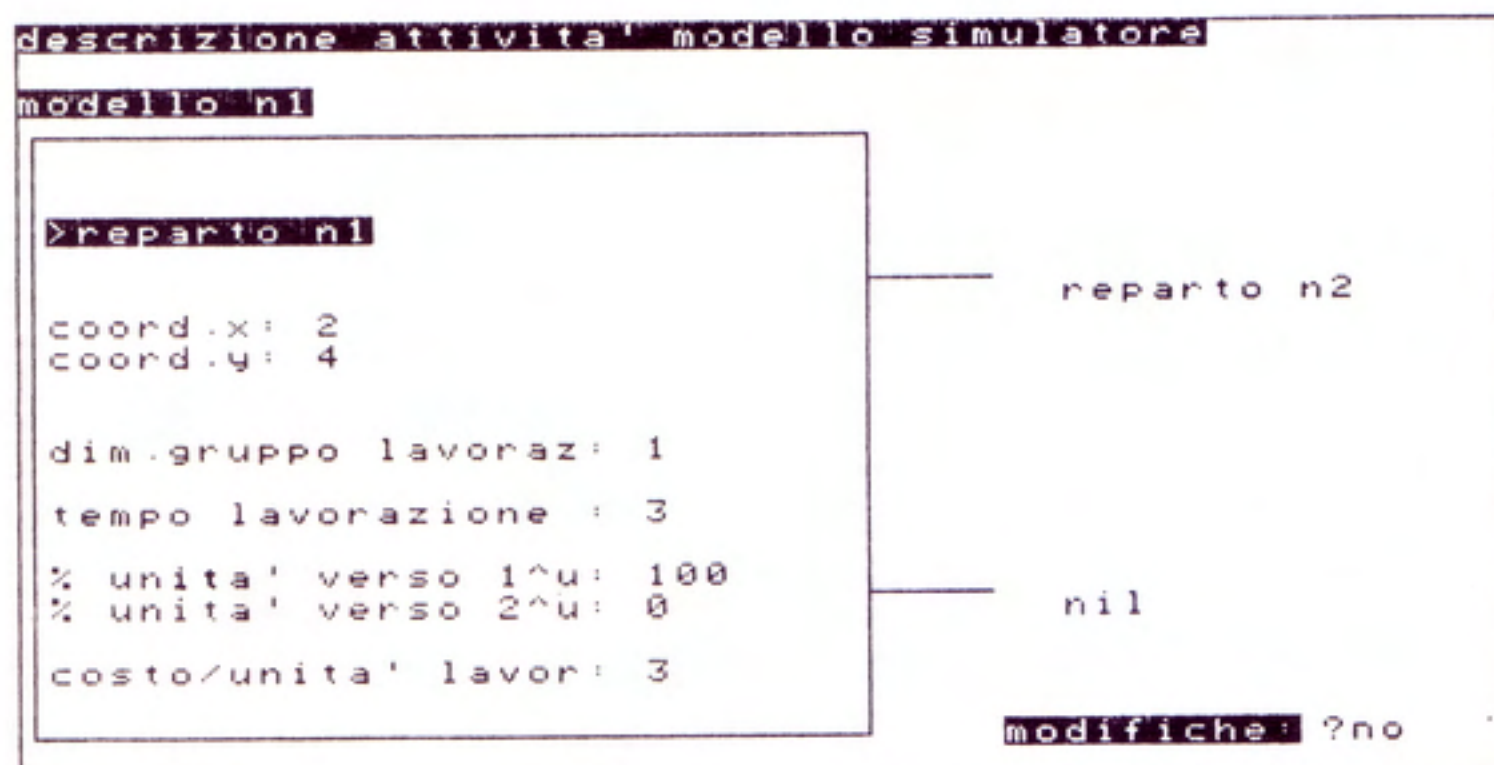



Figure 3 e 4 - Esempi di descrizione delle attività. Il reparto n. 1 è una sorgente (la sua prima lettera è >) e fornisce un pezzo alla volta in tre unità di tempo. Il reparto n. 2 lavora 2 pezzi per volta in 4 unità di tempo. Una è inviata al reparto n. 3 e una al reparto n. 4.

Dopo aver inserito informazioni sull'ottavo reparto, Forger rispose con un "no" alla richiesta:

+ATT (SI/NO)?

Il programma registrò le informazioni su disco e tornò al menu.

La key #2

Forger usò la key #2 per ottenere un grafico al video e alla stampante del modello che aveva appena creato.

La key #3

Con questa key Forger informò il programma sulla lunghezza del periodo da simulare.

Al video compare una rappresentazione grafica del primo reparto con il tempo corrente, la produzione totale, il costo e il tempo di inattività per mancanza di materiale da lavorare.

Durante la simulazione si servì di tre key per:

- key #1: ottenere risultati statistici parziali (figura 5);
- key #2: indicare il reparto da seguire al video;
- key #3: ottenere la stampa del grafico (figura 6).

```
tabella riassuntiva risultati simulazione
==========================================

tempo simulazione: 50
```

descrizione	t.input	f.input	mx.input	t.produz	t.costi	attesa inp
>reparto n1	0	0	0	16	48	0
reparto n2	16	0	1	16	64	12
reparto n3	7	3	3	4	8	47
reparto n4	7	0	0	7	28	22
reparto n5	2	0	1	2	6	42
reparto n6	9	1	2	8	32	34
reparto n7	6	2	3	4	20	26
reparto n8	8	0	1	8	24	37

```
tabella riassuntiva risultati simulazione
==========================================

tempo simulazione: 56
```

descrizione	t.input	f.input	mx.input	t.produz	t.costi	attesa inp
>reparto n1	0	0	0	18	54	0
reparto n2	18	0	1	18	72	13
reparto n3	8	0	3	8	16	50
reparto n4	8	0	0	8	32	24
reparto n5	4	1	1	3	9	45
reparto n6	12	2	2	10	40	37
reparto n7	6	1	3	5	25	26
reparto n8	9	1	1	8	24	40

Figura 5 - Sommario statistico dopo 50 e 56 unità temporali (e qualche minuto di elaborazione).
Le colonne indicano: totale pezzi provenienti da altre attività, lunghezza della coda alla 50ª un. di tempo, max. lunghezza della coda osservata, produzione totale, costo totale, tempo di attesa pezzi.

```
simulazione in corso
modello n1                          tempo   7

                >reparto n1
                prod.tot:  2                 2
                cost.tot:  6                 0
   0            t.attesa:  0
   0
                                             0
```

```
simulazione in corso
modello n1                          tempo   39

                reparto n2
                prod.tot:  12                6
                cost.tot:  48                1
   13           t.attesa:  10
   1
                                             6
                                             0
```

Figura 6 - Istantanee nel corso della simulazione, ottenute con la key 3. In entrata ed in uscita le cifre in "reverse" rappresentano i totali in ingresso ed in uscita fino al quel tempo, le cifre sottostanti i valori istantanei.

tabella riassuntiva risultati simulazione

tempo simulazione: 101

descrizione	t.input	f.input	mx.input	t.produz	t.costi	attesa inp
>reparto n1	0	0	0	33	99	0
reparto n2	33	1	1	32	128	20
reparto n3	16	0	3	16	32	90
reparto n4	16	0	0	16	64	39
reparto n5	6	0	1	6	18	76
reparto n6	21	1	2	20	80	60
reparto n7	16	3	5	13	65	26
reparto n8	22	0	1	22	66	57

Figura 7 - Tabella risultati statistici alla fine della simulazione. Per una descrizione delle colonne vedi figura 5.

n^	descrizione	x	y	d.grup	t.lavor	%out	cost/un	attivita'uscit.
1	>reparto n1	2	4	1	3	100	3	reparto n2
						0		nil
2	reparto n2	4	4	2	4	50	4	reparto n3
						50		reparto n4
3	reparto n3	6	2	4	2	50	2	reparto n5
						50		reparto n6
4	reparto n4	6	6	1	3	100	4	reparto n6
						0		nil
5	reparto n5	8	2	1	3	100	3	reparto n7
						0		nil
6	reparto n6	9	4	2	3	50	4	reparto n7
						50		reparto n8
7	reparto n7	11	2	1	5	100	5	reparto n8
						0		nil
8	reparto n8	15	4	2	3	0	3	nil
						0		nil

Figura 8 - Descrizione parametri attività. Nell'ordine: descrizione letterale (fino a 15 lettere), coordinate nella matrice 17★7, n. pezzi lavorati simultaneamente, tempo di lavorazione, % pezzi verso le due uscite, costo per pezzo, descrizione delle attività che ricevono i pezzi lavorati.

Dopo 30 minuti la simulazione era terminata. Forger premette la key #4 ed ottenne un tabulato dei risultati statistici (figura 7).

La key #5

Si servì di questa key per ottenere alla stampante una tabella dei parametri delle attività così come le aveva impostate (figura 8).

La key #7

Questa key lo deluse, perché ne risultò solo un cortese messaggio di commiato.

La morale

Forger raccolse gli stampati, spense il computer e chiamò il suo autista.

Nella riunione del giorno successivo informò il consiglio della decisione che aveva preso precisando che essa era stata dettata dalla sua decennale esperienza e fiuto manageriale. Naturalmente si "dimenticò" del piccolo aiuto ottenuto dal suo HP e trattò con la solita indifferenza i suoi collaboratori del gruppo dei modelli matematici.

Morale:

"Analisti, non fidatevi dei vostri computer; essi tramano alle vostre spalle e, quando sono spenti, sognano una buona occasione per pugnalarvi".

Conclusioni e note

Il programma è scritto in BASIC, ma è organizzato in segmenti e procedure e non contiene GOTO numerici (non esiste antidoto al contagio Pascal... per fortuna).

La figura 9 riassume la struttura del programma per i livelli delle sue procedure e le figure 10, 11, 12, 13, la sua rappresentazione grafica con le d-chart per il primo livello descrittivo.

Come i diagrammi di struttura, le d-chart rappresentano un miglioramento rispetto ai flow chart, e mostrano, a mio parere un vantaggio non tra-

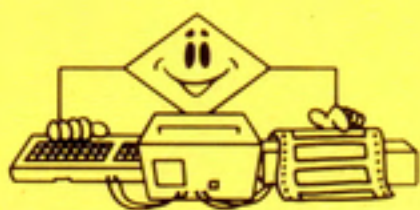



Figura 9 - Riassunto della struttura del programma e delle sue procedure.

MODELLI SIMULATORI

Variabili | Binari | Menu

FRE-MODELLO

Nuovo-file | Lettura-file | Al-video

Modifiche | Nuove-attivita' | Archiviazione

GRAF-MODELLO

SIMULAZIONE

Risultati-p

Inp-attivita'

Attivita'-da-seguire

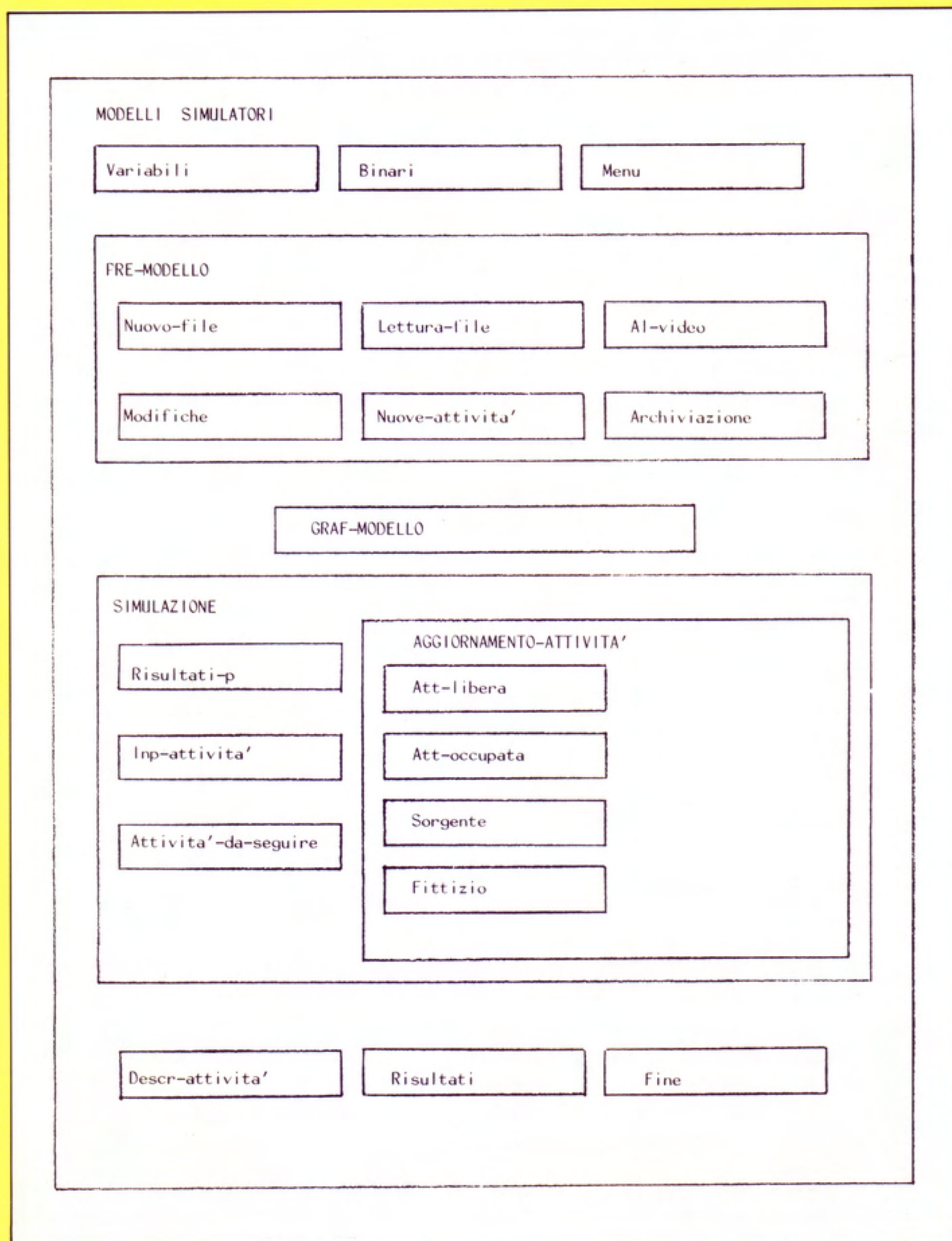
AGGIORNAMENTO-ATTIVITA'

Att-libera

Att-occupata

Sorgente

Fittizio

Descr-attivita' | Risultati | Fine

scurabile. Con le d-chart infatti la traduzione di un programma in BASIC o in Pascal è più semplice, perché questo metodo si basa su uno sviluppo verticale delle procedure dello stesso livello. Nei diagrammi di struttura questo sviluppo è invece orizzontale.
Riporto di seguito alcune regole per la corretta interpretazione delle d-chart, ripromettendomi, se **Bit** lo riterrà opportuno, di tornare sull'argomento con un articolo specifico:

- la lettura procede dall'alto fino ad un nodo indicato con un quadratino o un rombo che racchiudono una lettera;
- in corrispondenza di un nodo si scende a destra fino al prossimo nodo. Se il nodo era rappresentato da un rombo si procede a destra solo se la condizione indicata sul ramo del rombo è vera;
- in fondo alla diramazione di un nodo vi è sempre un triangolo con una lettera. Se il vertice del triangolo è rivolto verso il basso si torna al nodo

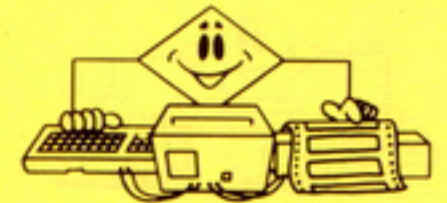



Figure 10, 11, 12, 13 - D-chart del primo livello descrittivo.

MODELLI SIMULATORI

VARIABILI

BINARI

MENU

repeat until
KEY 7

a

FINE

repeat until keypress

b

RICHIESTA KEY

b

c

case

key 1

PREPARAZIONE MODELLO

key 2

RAPPRESENTAZIONE GRAF.

key 3

SIMULAZIONE

key 4

RISULTATI SIMULAZIONE

key 5

DESCR.ATTIVITA'

a

c

Figura 11.

PRE-MODELLO
INPUT MODELLO ?
d
il file non esiste
NUOVO - FILE
d
LETTURA - FILE
e
for i= 1 to 119
NUOVE - ATTIVITA'
ARCHIVIAZIONE
MENU
c
f
attivita$(i)≠ "*"
AL - VIDEO
MODIFICHE
e
f

indicato dalla lettera e si esegue l'istruzione sotto il nodo.
Se il vertice del triangolo è rivolto verso l'alto siamo in presenza di un loop che va eseguito fino a che la condizione sul nodo di diramazione non è più vera o fino a che il FOR...NEXT è completato.

Ancora qualche nota per un utilizzo ottimale. Il programma utilizza delle istruzioni speciali non previste nel BASIC residente, ma che sono caricate in memoria con i binari "UTIL/1" e "GDUMP" inclusi nel disco sistema fornito dalla HP. Questi programmi devono essere dunque presenti sul disco che contiene il programma di simulazione.
È possibile definire fino a 119 attività (reparti) con una matrice 17★7 come indicato nella figura 2.
Le attività sorgenti non necessitano di un input per generare pezzi lavorati ed indicano per lo più i fornitori. La 1ª lettera della loro descrizione deve essere ">" (esempio >Reparto n1).
Non sono previste più di due uscite per attività, ma questa limitazione viene superata con i blocchi "fittizi" come indicato nella figura 10. Queste attività/fittizie hanno come prima lettera uno zero (esempio ORepfittizio).

Figura 12

GRAF_MODELLO

Intestazioni
& matrice graf

g
for i=1 to 119

h
attività$(i)≠ "*"

disegno attività
e defin. uscite

g

h

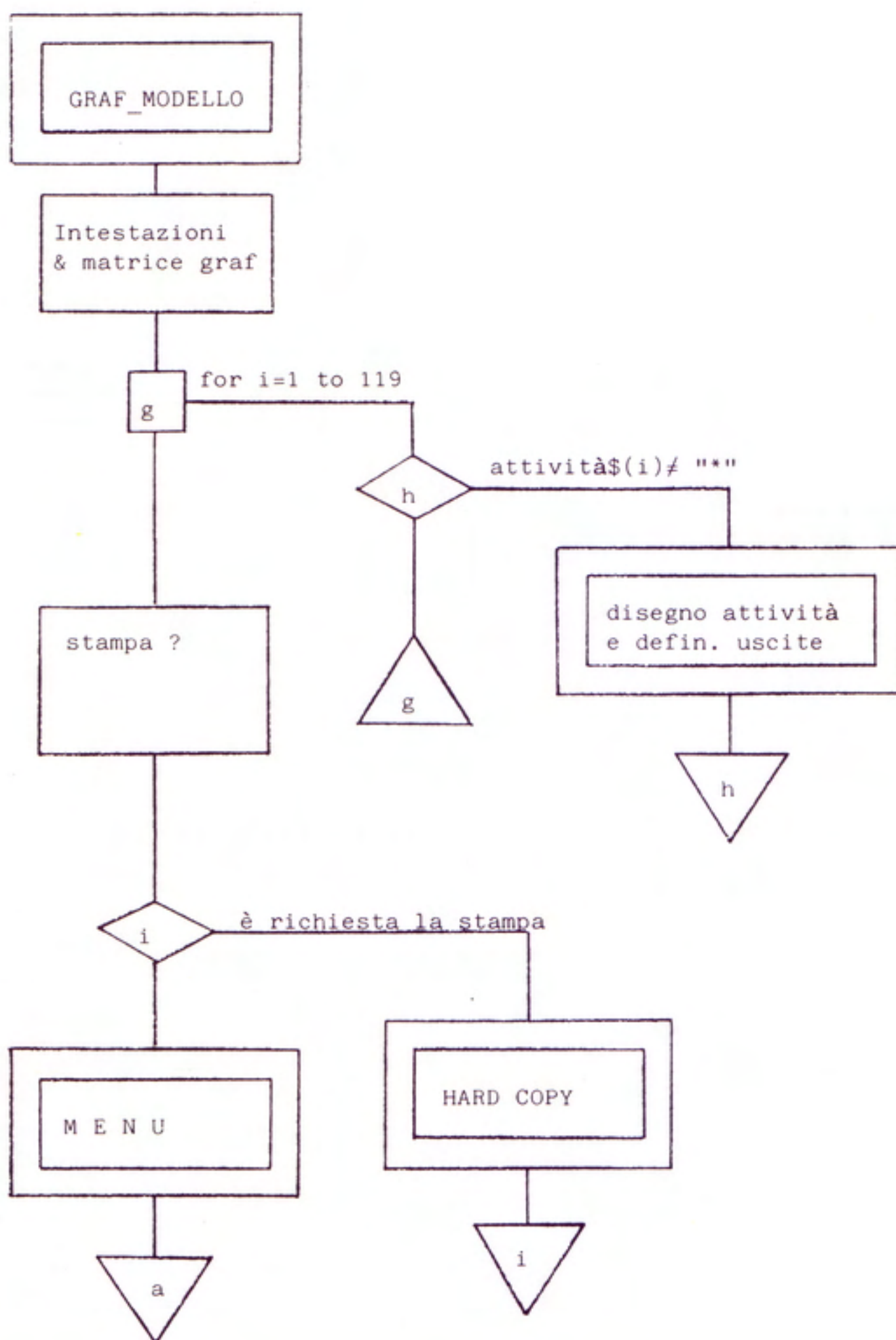
stampa ?

i
è richiesta la stampa

M E N U

HARD COPY

a

i

Le uscite vuote vanno indicate con la parola "Nil".

I parametri per ciascuna attività sono:

- coordinate x,y rispetto alla matrice grafica 17★7;
- numero pezzi lavorabili simultaneamente;
- tempo per la lavorazione dei pezzi alla voce precedente;
- % pezzi verso le due uscite. Una somma delle % maggiore di 100 indica una divisione di un pezzo in più parti e una somma minore di 100, un rendimento;
- costo per unità lavorata.

In fase di input, se il modello è presente su disco, il programma richiede le eventuali modifiche. Per correggere i parametri è sufficiente scrivere su quelli da modificare e passare con un "end line" sugli altri.

Per eliminare un'attività dal modello basta assegnarle il nome "★". Il programma considera costanti nel tempo le stime di produttività. Ho progettato una versione che lavori con distribuzioni statistiche, ma presentarla mi avrebbe costretto ad aprire in questa sede un argomento troppo specifico che avrebbe suscitato la critica di qualche dotto, e la noia di tutti gli altri.

Figura 13.

SIMULAZIONE
fori=0 to 119
azzeramento varia.
bili simulazione
c=n.attività prev.
for i = 1 to c
keypress
case key 1
RISULTATI ---P
2
INPUT- ATT.
3
DGRAPH
AGGIORNAMENTO
ATTIVITA'
in(att) > mxin(att)
mxin(att)=in(att)
M E N U

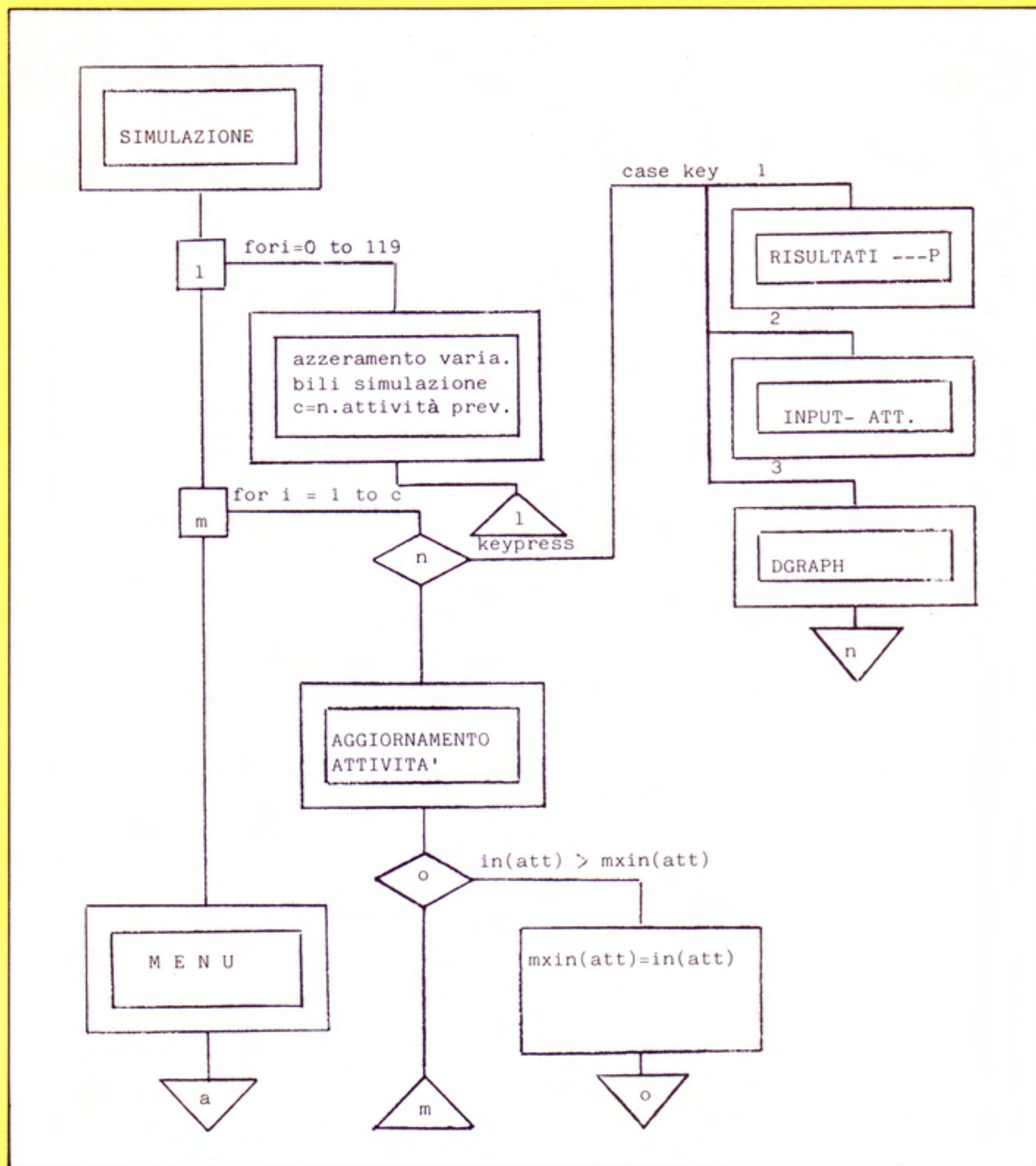
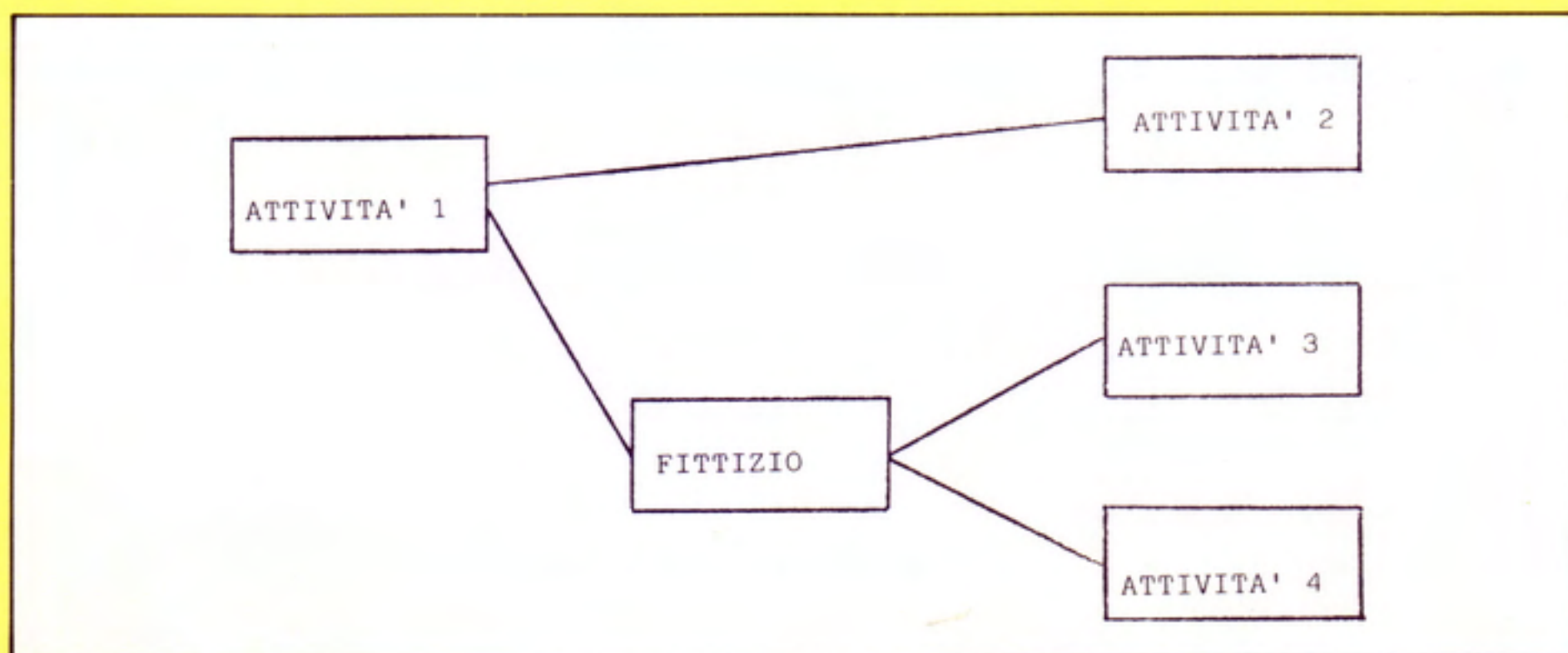



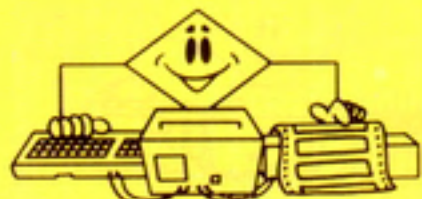
Figura 14 - Esempio di rappresentazione di attività a tre uscite, tramite introduzione di un blocco-fittizio. In generale il blocco-fittizio può puntare ad altri blocchi-fittizi, non esiste quindi limitazione teorica al numero di uscite possibili per attività.

In un altro programma, "Simulazione di un supermarket", ho considerato l'utilizzo della Poisson (in seguito ne considereremo la pubblicazione, Ndr).

Per concludere, un problema di ricerca operativa richiede una scelta oculata del modello e una conoscenza non superficiale delle tecniche più opportune... e questa sinceramente è la fase più delicata di un problema di questo tipo.

Comunque il computer rappresenta un utile strumento nella valutazione di alternative e nell'analisi delle code e efficienza di un progetto.

Listato 1 - Il programma descritto nell'articolo.

```
100 ! M O D E L L I   S I M U L A T O R I
150 ! MOSCARELLA FRANCESCO
200 ! =====================
250 PRINTER IS 701 @ PAGESIZE 24
300 GOSUB variabili
350 GOSUB binari
400 ON KEY# 1,"prep model" GOSUB pre_modello
450 ON KEY# 2,"graf model" GOSUB graf_modello
500 ON KEY# 3,"simulazion" GOSUB simulazione
550 ON KEY# 4,"risultati " GOSUB risultati
600 ON KEY# 5,"descr.att." GOSUB descr_att
650 ON KEY# 7,"fine progr" GOSUB fine
700 GOSUB menu
750 attesa_key: GOTO attesa_key
800 variabili: DIM attivita$(120)[15],p$(120,7)[15],a$(7)[40],yn$[25],l$[10]
850            DIM tempouscita(119),occupata(119),pezzi(119),a(119),pezzit(119)
900            DIM usc(120,2),in(119),tin(119),d$[80],attp(119),mxin(119)
950            a$(1)="coord.x:"
1000            a$(2)="coord.y:"
1050            a$(3)="dim.gruppo lavoraz:"
1100           a$(4)="tempo lavorazione :"
1150           a$(5)="% unita' verso 1^u:"
1200           a$(6)="% unita' verso 2^u:"
1250           a$(7)="costo/unita' lavor:"
1300           v$="*" @ flagpre_modello=0 @ flaggraf_modello=0
1350 RETURN
1400 binari:
1450           ON ERROR GOTO no_errore
1500           LOADBIN "UTIL/1" @ LOADBIN "GDUMP"
1550 no_errore: OFF ERROR
1600 RETURN
1650 menu:
1700           CLEAR @ KEY LABEL
1750           DISP HGL?$ ("programma di simulazione a blocchi",1) @ DISP
1800           DISP "key#1:preparazione modello simulatore"
1850           DISP "key#2:rappresentazione grafica del modello" @ DISP
1900           DISP "key#3:simulazione" @ DISP
1950           DISP "key#4:risultati statistici della simulazione"
2000           DISP "key#5:descrizione attivita'" @ DISP
2050           DISP "key#7:fine del programma"
2100 RETURN
2150 ! ================================================================
2200 pre_modello:
2250           c=0 @ PLOTTER IS 1
2300           CLEAR @ DISP HGL?$ ("preparazione del modello",1) @ DISP
2350           DISP "indica con > come prima lettera le att.sorgenti"
2400           DISP "indica con 0 come prima lettera le att.fittizie"
2450           DISP "indica con nil le uscite non definite"
2500           DISP "indica con * le attivita' che vuoi eliminare"
2550           DISP "modello:" @ INPUT modello$
2600           ON ERROR GOSUB nuovo_file
2650           ASSIGN# 1 TO  modello$
2700           OFF ERROR
2750           GOSUB lettura_file
2800           CLEAR @ DISP "modifiche:" @ INPUT yn$
2850           IF yn$="no" THEN new
2900           FOR i=1 TO 119
2950            IF attivita$(i) <> "*" THEN GOSUB al_video  @ GOSUB modifiche
3000           NEXT i
3050 new:
3100           GOSUB nuove_attivita
3150           GOSUB archiviazione
3200           GOSUB menu
3250           flagpre_modello=1
3300 RETURN
3350 nuovo_file:
3400           OFF ERROR
3450           CREATE modello$,120,120
3500           ASSIGN# 1 TO  modello$
3550           FOR i=1 TO 120
3600            PRINT# 1 ; v$
3650            FOR l=1 TO 7
3700             PRINT# 1 ; v$
3750            NEXT l
3800           NEXT i
```

HP

Seguito listato 1.

```
3850           ASSIGN# 1 TO  * @ ASSIGN# 1 TO  modello$
3900 RETURN
3950 lettura_file:
4000           FOR i=1 TO 120
4050            READ# 1 ; attivita$(i)
4100            FOR l=1 TO 7
4150             READ# 1 ; p$(i,l)
4200            NEXT l
4250           NEXT i
4300           ASSIGN# 1 TO  *
4350 RETURN
4400 al_video:
4450           PLOTTER IS 1 @ GCLEAR @ FRAME
4500           SCALE 0,49,239,0
4550           MOVE .5,225 @ DRAW .5,35 @ DRAW 28,35 @ DRAW 28,225
4600           DRAW .5,225 @ MOVE 28,80 @ DRAW 32,80 @ MOVE 28,180 @ DRAW 32,180

4650           FAST LABEL 0,0,"descrizione attivita' modello simulatore",1
4700           FAST LABEL 0,20,modello$,1
4750           FAST LABEL 1,60,attivita$(i),1
4800           y=i MOD 7 @ x=INT ((i-1)/7)+1
4850           d$=a$(1)&" "&VAL$ (x) @ IF i=120 THEN d$=a$(1)
4900           FAST LABEL 1,90,d$,0
4950           d$=a$(2)&" "&VAL$ (y) @ IF i=120 THEN d$=a$(2)
5000           FAST LABEL 1,100,d$,0
5050           d$=a$(3)&" "&p$(i,1) @ FAST LABEL 1,130,d$,0
5100           d$=a$(4)&" "&p$(i,2) @ FAST LABEL 1,150,d$,0
5150           d$=a$(5)&" "&p$(i,3) @ FAST LABEL 1,170,d$,0
5200           d$=a$(6)&" "&p$(i,4) @ FAST LABEL 1,180,d$,0
5250           d$=a$(7)&" "&p$(i,5) @ FAST LABEL 1,200,d$,0
5300           FAST LABEL 35,80,p$(i,6),0 @ FAST LABEL 35,180,p$(i,7),0
5350 RETURN
5400 modifiche:
5450           FAST LABEL 33,220,"modifiche:",1
5500           MOVE 43,227 @ INPUT yn$
5550           IF yn$="no" THEN GOTO stampa
5600 nuove:
5650           SGCLEAR 8,220,170,180
5700           MOVE 1,67 @ INPUT yn$@ IF yn$ <> "" THEN attivita$(i)=yn$
5750           SGCLEAR 160,220,100,110 @ l=1 @ x=21 @ y=137 @ GOSUB modif
5800           SGCLEAR 160,220,80,90 @ l=2 @ y=157 @ GOSUB modif
5850           SGCLEAR 160,220,50,70 @ l=3 @ y=177 @ GOSUB modif
5900           l=4 @ y=187 @ GOSUB modif
5950           SGCLEAR 160,220,30,40 @ l=5 @ y=207 @ GOSUB modif
6000           SGCLEAR 270,380,150,160 @ l=6 @ x=35 @ y=87 @ GOSUB modif
6050           SGCLEAR 270,380,50,60 @ l=7 @ y=187 @ GOSUB modif
6100 stampa:
6150           CLEAR @ DISP "hard copy:" @ INPUT stampa$
6200           IF stampa$="si" THEN DUMP GRAPHICS
6250 RETURN
6300 modif:
6350           MOVE x,y @ INPUT yn$@ IF yn$ <> "" THEN p$(i,l)=yn$
6400 RETURN
6450 nuove_attivita:
6500           i=120 @ GOSUB al_video
6550           FAST LABEL 30,220,"+att.(si/no)",1 @ MOVE 42,227 @ INPUT yn$
6600           IF yn$="no" THEN RETURN
6650           SGCLEAR 80,110,130,160
6700           MOVE 11,97 @ INPUT x@ MOVE 11,107 @ INPUT y@ i=(x-1)*7+y
6750           IF i<1 OR i>119 THEN i=120 @ GOTO nuove_attivita
6800           GOSUB nuove
6850           GOTO nuove_attivita
6900 RETURN
6950 archiviazione:
7000           ASSIGN# 1 TO  modello$
7050           FOR i=1 TO 120
7100            PRINT# 1 ; attivita$(i)
7150            FOR l=1 TO 7
7200             PRINT# 1 ; p$(i,l)
7250            NEXT l
7300           NEXT i
7350           ASSIGN# 1 TO  *
7400 RETURN
7450 ! ===================================================================
7500 graf_modello: IF NOT flagpre_modello THEN RETURN
```

Seguito listato 1.

```
7550            PLOTTER IS 1 @ GCLEAR @ FRAME
7600            FAST LABEL 0,0,"grafo del modello simulatore",1
7650            FAST LABEL 33,0,"batti cont",0
7700            FAST LABEL 0,15,modello$,1
7750            LOCATE 20,150,15,80 @ FRAME @ SCALE 1,18,1,8 @ LINE TYPE 3
7800            GRID 1,1,1,1,1,1 @ LINE TYPE 1
7850            FOR i=1 TO 7
7900             d$=VAL$ (i) @ MOVE 0,i+.5 @ LABEL d$ @ MOVE 18.7,i+.5 @ LABEL d$

7950            NEXT i
8000            DEG @ LDIR 270
8050            FOR i=1 TO 17
8100             d$=VAL$ (i) @ MOVE i+.5,.3 @ LABEL d$ @ MOVE i+.5,8.7 @ LABEL d$

8150            NEXT i
8200            LDIR 0 @ k=0
8250            FOR i=1 TO 119
8300             IF attivita$(i)="*" THEN non_disegna
8350             k=k+1
8400             x=INT ((i-1)/7)+1 @ y=i MOD 7
8450             MOVE x,y @ DRAW x,y+1 @ DRAW x+1,y+1 @ DRAW x+1,y @ DRAW x,y
8500            MOVE x+.1,y+.1 @ LABEL VAL$ (k)
8550             FOR z=1 TO 2
8600              FOR l=i+1 TO 119
8650               IF p$(i,z+5) <> attivita$(l) THEN non_trovato
8700               x(z)=INT ((l-1)/7)+1 @ y(z)=l MOD 7
8750               usc(i,z)=1
8800               MOVE x+1,y+.5 @ DRAW x(z),y(z)+.5
8850               GOTO altra_uscita
8900 non_trovato:
8950              NEXT l
9000              usc(i,z)=0
9050 altra_uscita:
9100             NEXT z
9150 non_disegna:
9200            NEXT i
9250            flaggraf_modello=1 @ PAUSE
9300            CLEAR @ DISP "stampa del grafico:" @ INPUT stampa$
9350            IF stampa$="si" THEN DUMP GRAPHICS
9400            GOSUB menu
9450 RETURN
9500 ! ====================================================================
9550 simulazione:
9600            IF NOT flaggraf_modello THEN RETURN
9650            CLEAR
9700            DISP @ DISP "programma simulatore a blocchi"
9750            DISP @ DISP modello$ @ DISP
9800            DISP "durante la simulazione:"
9850            DISP ":::::::::::::::::::::::"
9900            DISP "pause:sospende la simulazione"
9950            DISP "cont :riprende la simulazione"
10000            DISP "key#1:risultati parziali"
10050            DISP "key#2:input attivita' da seguire"
10100            DISP "key#3:stampe durante la simulazione"
10150            DISP @ DISP
10200            DISP "stampa risultati parziali:" @ INPUT stampa$
10250            DISP "tempo totale simulazione:" @ INPUT tempot
10300            PLOTTER IS 1 @ GCLEAR @ FRAME @ c=0 @ dg=0
10350            SCALE 0,49,239,0
10400           FAST LABEL 0,0,"simulazione in corso",1
10450           FAST LABEL 0,20,modello$,1
10500           MOVE 13,120 @ DRAW 15,120 @ DRAW 15,40 @ DRAW 35,40
10550           DRAW 35,200 @ DRAW 15,200 @ DRAW 15,120
10600           MOVE 35,160 @ DRAW 37,160 @ MOVE 35,80 @ DRAW 37,80
10650           FAST LABEL 17,80,"prod.tot:",0
10700           FAST LABEL 17,100,"cost.tot:",0
10750           FAST LABEL 17,120,"t.attesa:",0
10800           FAST LABEL 29,20,"tempo   :",0 @ attivita$(0)="*"
10850           FOR i=0 TO 119
10900            tempouscita(i)=0 @ occupata(i)=0 @ in(i)=0
10950            pezzit(i)=0 @ attp(i)=0 @ tin(i)=0 @ mxin(i)=0
11000            IF attivita$(i)[1,1]="*" THEN nexti
```

HP

Seguito listato 1.

```
11050          c=c+1 @ a(c)=i
11100          IF attivita$(i)[1,1]=">" THEN tempouscita(i)=VAL (p$(i,2))
11150 nexti:
11200         NEXT i
11250         natts=a(1)
11300         ON KEY# 1 GOSUB risultati_p
11350         ON KEY# 2 GOSUB inp_att
11400         ON KEY# 3 GOSUB dgraph
11450         FOR tempo=1 TO tempot
11500          FOR i=1 TO c
11550           natt=a(i) @ GOSUB aggiorn_attivita
11600           IF in(natt)>mxin(natt) THEN mxin(natt)=in(natt)
11650          NEXT i
11700          GOSUB attivita_da_seguire
11750          IF dg THEN DUMP GRAPHICS @ dg=0
11800         NEXT tempo
11810         FAST LABEL 33,0,"batti cont",0
11850         ON KEY# 1,"prep model" GOSUB pre_modello
11900         ON KEY# 2,"graf model" GOSUB graf_modello
11950         ON KEY# 3,"simulazion" GOSUB simulazione
12050         PAUSE @ GOSUB menu
12100 RETURN
12150 inp_att:
12200         CLEAR
12250         DISP "attivita' da seguire:" @ INPUT yn$
12300         FOR l=1 TO c
12350          IF attivita$(a(l))=yn$ THEN natts=a(l)
12400         NEXT l
12450 RETURN
12500 aggiorn_attivita:
12550         pezzi(natt)=VAL (p$(natt,1))
12600         vers1=INT (pezzi(natt)*VAL (p$(natt,3))/100+.5)
12650         vers2=INT (pezzi(natt)*VAL (p$(natt,4))/100+.5)
12700         IF attivita$(natt)[1,1]="0" THEN fittizio
12750         IF attivita$(natt)[1,1]=">" THEN sorgente
12800         IF occupata(natt) THEN att_occupata
12850 att_libera:
12900         tlavor=VAL (p$(natt,2))
12950         IF in(natt)<pezzi(natt) THEN attp(natt)=attp(natt)+1 @ RETURN
13000         in(natt)=in(natt)-pezzi(natt)
13050         pezzit(natt)=pezzit(natt)+pezzi(natt)
13100         occupata(natt)=1
13150         tempouscita(natt)=tempo+tlavor
13200 RETURN
13250 att_occupata:
13300         IF tempo<tempouscita(natt) THEN RETURN
13350         in(usc(natt,1))=in(usc(natt,1))+vers1
13400         in(usc(natt,2))=in(usc(natt,2))+vers2
13450         tin(usc(natt,1))=tin(usc(natt,1))+vers1
13500         tin(usc(natt,2))=tin(usc(natt,2))+vers2
13550         occupata(natt)=0
13600 RETURN
13650 fittizio:
13700         vers1=INT (in(natt)*VAL (p$(natt,3))/100+.5)
13750         vers2=INT (in(natt)*VAL (p$(natt,4))/100+.5)
13800         in(usc(natt,1))=in(usc(natt,1))+vers1
13850         in(usc(natt,2))=in(usc(natt,2))+vers2
13900         in(natt)=0
13950         tin(usc(natt,1))=tin(usc(natt,1))+vers1
14000         tin(usc(natt,2))=tin(usc(natt,2))+vers2
14050 RETURN
14100 sorgente:
14150         IF tempo<tempouscita(natt) THEN RETURN
14200         pezzit(natt)=pezzit(natt)+pezzi(natt)
14250         tlavor=VAL (p$(natt,2))
14300         in(usc(natt,1))=in(usc(natt,1))+vers1
14350         in(usc(natt,2))=in(usc(natt,2))+vers2
14400         tin(usc(natt,1))=tin(usc(natt,1))+vers1
14450         tin(usc(natt,2))=tin(usc(natt,2))+vers2
14500         tempouscita(natt)=tempo+tlavor
14550 RETURN
14600 attivita_da_seguire:
```

Seguito listato 1.

```
14650          SGCLEAR 290,380,35,220 @ SGCLEAR 10,110,90,140
14700          SGCLEAR 210,280,40,170 @ SGCLEAR 140,280,160,180
14750          FAST LABEL 17,60,attivita$(natts),1
14800          d$=VAL$ (tempo) @ FAST LABEL 37,20,d$,1
14850          d$=VAL$ (tin(natts)) @ FAST LABEL 2,110,d$,1
14900          d$=VAL$ (in(natts)) @ FAST LABEL 2,130,d$,0
14950          d$=VAL$ (pezzit(natts)) @ FAST LABEL 27,80,d$,1
15000          costo=pezzit(natts)*VAL (p$(natts,5))
15050          d$=VAL$ (costo) @ FAST LABEL 27,100,d$,1
15100          d$=VAL$ (attp(natts)) @ FAST LABEL 27,120,d$,1
15150          t=INT (pezzit(natts)*VAL (p$(natts,3))/100)
15200          d$=VAL$ (t) @ FAST LABEL 38,80,d$,1
15250          t=INT (pezzit(natts)*VAL (p$(natts,4))/100)
15300          d$=VAL$ (t) @ FAST LABEL 38,160,d$,1
15350          d$=VAL$ (in(usc(natts,1)))
15400          IF usc(natts,1)<> 0 THEN FAST LABEL 38,100,d$,0
15450          d$=VAL$ (in(usc(natt,2)))
15500          IF usc(natts,2)<> 0 THEN FAST LABEL 38,180,d$,0
15550 RETURN
15600 ! ===================================================================
15650 risultati:
15700          CLEAR @ DISP "stampa risultati:" @ INPUT stampa$
15750 risultati_p:
15800          CLEAR @ IF stampa$="si" THEN CRT IS 701
15850          DISP "tabella riassuntiva risultati simulazione"
15900          DISP "========================================="
15950          DISP @ DISP "tempo simulazione:";tempo @ DISP
16000          DISP "descrizione";TAB (19);"t.input    ";
16050          DISP "f.input   mx.input  t.produz   ";
16100          DISP "t.costi   attesa inp" @ DISP
16150          FOR j=1 TO c
16200           d$=attivita$(a(j))
16250           x1=tin(a(j)) @ x2=in(a(j)) @ x3=mxin(a(j))
16300           x4=pezzit(a(j)) @ x5=x4*VAL (p$(a(j),5))
16350           x6=attp(a(j))
16400           DISP USING formato ; d$,x1,x2,x3,x4,x5,x6
16450          NEXT j
16500          DISP
16550          CRT IS 1 @ DISP "batti cont" @ PAUSE
16600          IF tempo>tempot THEN GOSUB menu
16650 formato: IMAGE 16A,6(9D,1X)
16700 RETURN
16750 ! ===================================================================
16800 descr_att:
16850          IF NOT flagpre_modello THEN RETURN
16900          PRINTER IS 701 @ k=0
16950          PRINT "n^   descrizione      x    y     ";
17000          PRINT " d.grup    t.lavor   %out cost/";
17050          PRINT "un  attivita'uscit."
17100          PRINT
17150          l$=".........."
17200          FOR i=1 TO 119
17250           IF attivita$(i)="*" THEN no_stampa
17300          k=k+1
17350           x=INT ((i-1)/7)+1 @ y=i MOD 7
17400          PRINT k;TAB (6);attivita$(i);TAB (22);x;TAB (26);y;
17450          PRINT TAB (32);p$(i,1);TAB (41);p$(i,2);TAB (51);p$(i,3);
17500          PRINT TAB (56);p$(i,5);TAB (65);p$(i,6)
17550          PRINT TAB (51);p$(i,4);TAB (65);p$(i,7)
17600          PRINT l$;l$;l$;l$;l$;l$;l$;l$
17650 no_stampa:
17700          NEXT i
17750          GOSUB menu
17800 RETURN
17850 dgraph: dg=1 @ RETURN
17900 ! ===================================================================
17950 fine:
18000          CLEAR @ DISP @ DISP
18050          DISP HGL?$ ("fine del programma",1)
18100          DISP @ DISP "riferimento:bit n...."
18150          DISP "per informazioni: 0823/468725"
18200 END
```

Statistica a due dimensioni

Prosegue con questo numero la serie di articoli concernenti le applicazioni del calcolo statistico alla programmazione.
Proseguiamo con un programma che permette la rappresentazione grafica o sintetica di variabili statistiche a due dimensioni. Analogamente a quanto fatto per l'articolo precedente della stessa serie, abbiamo ritenuto opportuno premettere una breve trattazione degli argomenti toccati dall'articolo stesso.

di **Umberto Barzaghi**

Parte prima

La variabile statistica a due dimensioni

Consideriamo una popolazione i cui individui possano essere classificati secondo due caratteristiche possedute. Ciascuna delle due caratteristiche ha i quattro requisiti cui, come abbiamo visto nell'articolo precedente, devono soddisfare anche le variabili semplici, vale a dire:
- che l'attributo possa assumere forme Xi e Yj distinte tra loro incompatibili;
- che ogni forma Xi e Yj rappresenti un concetto di classe;
- che in ogni individuo della popolazione sia presente una delle forme Xi e Yj;
- che la popolazione non si presenti totalmente omogenea.

Se i requisiti suddetti sono soddisfatti, la variabile statistica costruita sulla popolazione data può essere considerata argomento di classificazione statistica; tenendo conto della duplice modalità degli individui, viene chiamata variabile statistica doppia o a due dimensioni.
Ogni individuo della popolazione viene quindi classificato in base ad una duplice successione di valori argomentali (X,Y), costituendo quindi una tabella a doppia entrata, che rappresenta numericamente la variabile statistica a due dimensioni.

Supponendo di aver creato m classi per quanto riguarda il primo argomento ed n per il secondo, otterremo due serie di valori (x1, x2,..., xm) e (y1, y2,..., yn) che gli argomenti X e Y possono assumere e che costituiscono l'elemento metrico della tabella; mentre gli mxn valori di f(i,j), cioè gli mxn valori contenuti nelle caselle della tabella, rappresentano l'elemento connettivo.

Ogni valore f(i,j) rappresenta la frequenza o numero di individui della popolazione in possesso della coppia (xj, yi) di valori argomentali appartenenti alle serie X e Y. Ovviamente la sommatoria doppia per i da 1 a n e per j da 1 a m di f(i,j) deve dare come risultato la numerosità N della popolazione, cioè il numero di campioni esaminati.
L'ultima riga e l'ultima colonna della tabella contengono rispettivamente:
pj = sommatoria da 1 a n di f(i,j), frequenza con cui la X assume il valore xj, qualunque sia il valore assunto dalla Y;
qi = sommatoria da 1 a m di f(i,j), frequenza con cui la Y assume il valore yi, qualunque sia il valore assunto dalla X.
Le qi sono le frequenze della variabile statistica semplice Y (a una dimensione) ottenuta analizzando la popolazione solamente sotto la caratteristica Y. Viceversa le pj sono le frequenze della variabile statistica semplice X.
La media e la varianza generale della popolazione dal punto di vista dell'argomento Y indipendentemente da X, sono date, rispettivamente, dalla sommatoria da 1 a n del prodotto di yi per il rapporto tra la frequenza qi e la numerosità dei campioni raccolti N; e dalla sommatoria, sempre da 1 a n del quadrato della differenza tra yi e la media My per il rapporto suddetto. Formule corrispondenti, ovviamente, valgono per l'argomento X.
Ovviamente le sommatorie da 1 a m delle frequenze pj e da 1 a n delle qi devono dare come risultato la numerosità N dei campioni raccolti.

Curve di regressione e di variabilità

Della variabile statistica si possono dare anche rappresentazioni grafiche; uno dei modi più diffusi è quello chiamato del diagramma di dispersione, ottenuto rappresentando simbolicamente le frequenze f_{ij} delle coppie di valori (x_i, y_i) con cerchi di superficie proporzionale a f_{ij}, in corrispondenza dell'ascissa x_j e dell'ordinata y_i. Tuttavia tale metodo di rappresentazione, precluso ad ogni tipo di calcolatore privo di alta risoluzione, non permette valutazioni se non grossolanamente qualitative della struttura della popolazione. Assai più espressiva risulta la rappresentazione mediante due particolari curve ottenibili dopo una prima rielaborazione sintetica dei dati.

Così come la variabile statistica semplice si può sinteticamente riassumere, come abbiamo visto, in due indici, la media e la varianza, identicamente si può riassumere la tabella di figura 1 in due sole colonne che bastano, in generale, a caratterizzare la variabile a due dimensioni.
È necessario però, innanzitutto, stabilire, sia in base a considerazioni teoriche, che in base ad intuizione, quale dei due argomenti sia da considerare indipendente e quale invece dipendente dall'altro. Il primo verrà indicato, nel seguito, con la X ed il secondo con la Y. In molti casi la distin-

X \ Y	x_1	x_2	x_3	...	x_{m-1}	x_m	Totali per riga q_i
y_1	f_{11}	f_{12}	f_{13}	...	$f_{1(m-1)}$	f_{1m}	q_1
y_2	f_{21}	f_{22}	f_{23}	...	$f_{2(m-1)}$	f_{2m}	q_2
y_3	f_{31}	f_{32}	f_{33}	...	$f_{3(m-1)}$	f_{3m}	q_3
. . .	. . .	. . .	. . .		. . .	. . .	. . .
y_n	f_{n1}	f_{n2}	f_{N3}	...	$f_{n(m-1)}$	f_{nm}	q_n
Totali per colonna p_j	p_1	p_2	p_3	...	p_{m-1}	P_m	N

Figura 1 - Tabella a doppia entrata rappresentante una variabile statistica a due dimensioni.

zione è semplice; in tutti quei casi in cui, per esempio, uno dei due argomenti è una caratteristica ambientale (ora, temperatura, macchina operatrice...) che condiziona il risultato di una certa operazione senza esserne condizionata; in altri casi, invece, la scelta può risultare dubbia. In questi casi potranno essere convenienze pratiche a suggerire la scelta dell'argomento indipendente X e di quello dipendente Y.
Si consideri l'accoppiamento della prima colonna della tabella di figura 1 contenente i valori argomentali Y, con ciascuna delle successive colonne; si ottengono m variabili statistiche semplici, ciascuna caratterizzata da un diverso valore dell'argomento X posseduto dai propri individui.

Di ciascuna di queste variabili semplici si possono calcolare i due indici fondamentali, media e varianza.

Si otterrà pertanto una tabella sintetica ridotta a sole 4 colonne (vedi figura 2). La seconda e la terza colonna della tabella, prendono rispettivamente il nome di "media parziale per colonna" e "varianza parziale per colonna".
Questa tabella sintetica può essere rappresentata tramite due curve: la curva di regressione e la curva di variabilità nelle quali, rispettivamente, le medie M_{yi} e le varianze σ^2_{yi} vengono riportate sulla scala dei valori dell'argomento fondamentale X.

La costruzione delle due curve è assai utile per seguire l'andamento di un processo produttivo in serie; se il prodotto è costituito da un pezzo che deve possedere determinati requisiti, per esempio una dimensione prestabilita, si può esaminare regolarmente parte della produzione e, in funzione della data - variabile indipendente -, riportare in un grafico la media delle misure ottenute dalla verifica delle dimensioni dei pezzi e la varianza delle misure effettuate. La prima curva (curva di regressione) permette di evidenziare la presenza di eventuali fattori che perturbano sistematicamente la produzione, spostando progressivamente la dimensione media del pezzo. La seconda curva (curva di variabilità) permette di controllare l'eventuale insorgere nella produzione di cause perturbanti di carattere accidentale.

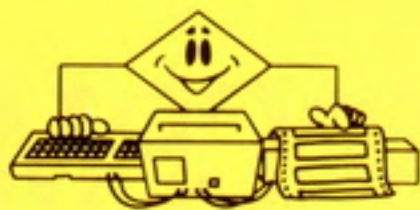
Misura della dipendenza fra gli argomenti di una variabile statistica doppia

È importante studiare non tanto le variabili X e Y indipendentemente, ma, piuttosto, sintetizzare con un indice quanto più significativo è possibile in quali relazioni stanno le due variabili simultaneamente presenti in un gruppo di individui.
È interessante, cioè, dare un criterio per giudicare se e quanto le due variabili siano reciprocamente dipendenti. Questa interdipendenza è facilmente rilevabile in fenomeni semplici e ripetibili; diventa assai problematico rilevarla quando il fenomeno si presenta complesso o nel caso in cui la dipendenza sia così tenue da poter essere mascherata da altre cause perturbanti.
L'indagine suddetta tende a stabilire se un individuo in possesso di un valore x_j dell'argomento X possa indifferentemente possedere qualsiasi valore dell'argomento Y o se, invece, la frequenza con cui assume i diversi valori di Y è dipendente dal valore dell'argomento X posseduto.

Introdurremo, perciò, un indice di dipendenza tra gli argomenti, che rappresenterà la misura della dipendenza stessa. Definiamo, innanzitutto,

X	M_{yj}	s^2_{yj}	p_j
x_1	M_{y1}	s^2_{y1}	p_1
x_2	M_{y2}	s^2_{y2}	p_2
.	.	.	.
.	.	. .	.
.	.	.	.
x_m	M_{ym}	s^2_{ym}	p_m

Figura 2 - Tabella sintetica ridotta: la seconda e la terza colonna contengono, rispettivamente, la media parziale per colonna e la varianza parziale per colonna.

i due casi limite, che non si verificheranno mai nei problemi reali, ma che sono, in via puramente teorica, mentalmente concepibili: i casi, cioè, di "perfetta dipendenza" e di "perfetta indipendenza" fra le due variabili x e y.
Si ha perfetta dipendenza quando il fatto che un individuo possegga un determinato valore x_j di X determina univocamente quale valore y_i di Y l'individuo stesso possiederà. In una tabella a doppia entrata, come quella di figura 1, ci sarà, quindi, in ogni colonna x_j una sola frequenza f_{ij} diversa da 0, in quanto tutti gli individui che presentano il valore argomentale x_j avranno tutti lo stesso valore argomentale y_i. Si può, cioè, supporre che esista una funzione $y'=f(x)$ che lega i due argomenti della variabile statistica doppia e che, di conseguenza, è detta curva di dipendenza.
L'equazione suddetta può essere determinata:
a) empiricamente: proponendo funzioni arbitrarie che si suppone si adattino bene alla descrizione della relazione esistente fra la x e la y;
b) razionalmente: creando, cioè, degli schemi logici che costituiscano una possibile spiegazione della relazione e non una semplice descrizione.
Nella maggior parte dei casi non si dispone né di curve di dipendenza del primo né del secondo

tipo. Si assume allora come curva di dipendenza la curva di regressione, precedentemente descritta. Si vedrà, in seguito, come essa rappresenti la più fedele descrizione dell'andamento della Y in funzione della X, ricavabile dai soli dati sperimentali.

Il comportamento delle due variabili, nel caso in cui esista fra di esse una perfetta indipendenza, può, a volte, essere ricavato dall'esperimento stesso, osservando, cioè, come si comporta la Y in assenza della X. Nella maggior parte dei casi, però, non è pratico o non è del tutto possibile analizzare la Y in assenza della X e si dovrà far ricorso ad ipotesi logicamente ammissibili. Precisamente, la statistica propone, come ipotesi di indipendenza in media o correlativa, quella in cui, in caso di indipendenza, le medie dei valori della Y (variabile dipendente) rimangono invariate per ogni valore x_j della X. Le medie parziali per colonna M_{yj} dovranno cioè essere tutte uguali fra loro ed uguali alla media generale M_y. Ciò significa considerare ogni colonna come provvista di una certa stabilità del valore medio. Si ha, perciò, perfetta indipendenza quando, avendo separato la popolazione in tanti gruppi in base ai valori dell'argomento X, ciascuno di essi può considerarsi come un gruppo di estrazioni a caso dalla popolazione presa in toto. Ogni gruppo avrà allora una media che oscillerà intorno al valore medio generale, con oscillazioni tanto minori quanto è maggiore il numero di individui in ciascun gruppo. Poiché l'aver scelto gli individui in base alle x non ha provocato una corrispondente scelta degli individui rispetto alle y, giacché essi si comportano come se fossero stati estratti a caso, si può dire che non vi è dipendenza tra X e Y.

Possiamo ora cercare di interpretare il valore argomentale y_i presente in un individuo qualunque fra quelli aventi valore argomentale x_j.

La media M_{yj} si può pensare come valore dominante nel gruppo suddetto, in quanto le eventuali differenze y_i-M_{yj} sono dovute a fattori accidentali o comunque indipendenti dalla variabile X, che si presenta con lo stesso valore in tutti gli individui del gruppo. Per questo motivo, le differenze y_i-M_{yj} sono dette scarti indipendenti e rappresentano le dispersioni dei singoli valori y_i intorno alla loro media, dovute a cause che non hanno nulla a che fare con l'influenza della X. Nel caso in cui si sia potuta fissare una relazione $y'=f(x)$ da assumere come curva di dipendenza, si dovrebbe avere in corrispondenza di ogni x_j una ben determinata y'_i ed anche qui le differenze y_i-y'_i sarebbero dovute a motivi accidentali o comunque indipendenti dalla x_j. Se, invece, analizziamo il significato della singola media per colonna M_{yj}, vediamo come essa si possa pensare scomposta in due parti, una indipendente dalla X, che esisterebbe, cioè, anche in assenza di essa, ed una causata da quel particolare valore x_j posseduto da tutti gli individui di cui M_{yj} è la media.

$$I_2 = \frac{\sum_j (y'_j - M_y)^2 p_j}{\sum_i \sum_j (y_i - y'_j)^2 f_{ij} + \sum_j (y'_j - M_y)^2 p_j}$$

$$\eta_x^2 = \frac{\sum_j (M_{yj} - M_y)^2 p_j}{\sum_i \sum_j (y_i - M_{yj})^2 f_{ij} + \sum_j (M_{yj} - M_y)^2 p_j}$$

Figura 3 - Indice quadratico di dipendenza ed indice di Pearson.

Supponiamo di considerare, ad esempio, il valore medio del frumento raccolto in un certo numero di campi a cui è stata somministrata una certa quantità x_j di fertilizzante; questo valore medio si compone di una parte che si sarebbe raccolta comunque, anche senza l'uso di fertilizzante, più una parte che è dovuta al fatto che il fertilizzante invece c'era, ed in misura x_j.

Poiché si è convenuto di considerare la media generale M_y come valore argomentale corrispondente all'ipotesi di indipendenza correlativa o in media, si potrà indicare la parte di M_{yj} che dipende da x_j, che prende il nome di scarto dipendente, come M_{yj}-M_y. L'attendibilità, o peso, di tale scarto, dipenderà ovviamente dal numero p_j di individui di cui si dispone per calcolare la media per colonna M_{yj}.

Avendo definito gli scarti dipendenti e quelli indipendenti, si può osservare che, se vi fosse tra la X e la Y una perfetta dipendenza, tutti gli individui con valore argomentale x_j avrebbero la stessa y_i, la quale coinciderebbe con la media M_{yj} e, quindi, tutti gli scarti indipendenti (y_i-M_{yj}) sarebbero nulli; mentre se avessero una perfetta indipendenza, tutte le medie parziali per colonna M_{yj} sarebbero uguali fra loro ed uguali alla media generale My, per cui tutti gli scarti dipendenti (M_{yj}-M_y) sarebbero nulli.

Un indice intuitivo che rappresenti il grado di dipendenza fra la X e la Y potrebbe essere il rapporto fra la sommatoria degli scarti dipendenti e la sommatoria degli scarti indipendenti; ma tale rapporto avrebbe il difetto di divenire infinito in caso di perfetta dipendenza. In pratica, si preferisce usare un indice quadratico di dipendenza che si può scrivere nelle due forme di figura 3. Nella seconda forma, l'indice quadratico di dipendenza prende il nome di Indice di Pearson ed è il più frequentemente usato. Va sottolineata la differenza fondamentale di significato esistente tra Indice di Pearson e I_2. Il primo misura il grado di dipendenza della Y dalla X, mentre il secondo misura quanto bene la particolare funzione $y'=f(x)$ usata per il calcolo dell'indice quadratico può rappresentare tale dipendenza. Sia I_2 che l'indice di Pearson sono uguali a 0 quando vi è perfetta indipendenza ed hanno valore compreso fra 0 e 1 in tutti gli altri casi. La dipendenza fra X e Y è da considerarsi nulla se il valore dell'indice è minore di 0,2; è presente, ma debole, quando è maggiore di 0,2 e minore di 0,4 e sensibile quando è compreso tra 0,4 e 0,6; forte quando va da 0,6 a 0,8 e fortissima se maggiore di 0,8.

Coefficienti di regressione

Si è visto, nel paragrafo precedente, che nel caso in cui esista una dipendenza tra i due argomenti X e Y di una variabile statistica doppia si può proporre una relazione analitica tra le due variabili. Quando, però, la dipendenza non è perfetta non è possibile trovare una funzione che faccia corrispondere esattamente ai valori argomentali x quelli corrispondenti y. Il problema viene quindi risolto con una funzione interpolatrice.

Particolare semplicità presenta il caso in cui la funzione interpolatrice $y'=f(x)$ può essere considerata una retta ed assumere la forma $y'=ax+b$. Se la dipendenza tra X e Y non è perfetta, i valori di y' differiscono dai corrispondenti y, qualunque sia la scelta dei coefficienti a e b; la differenza y'-y ha il significato di uno scarto, quindi si dovranno cercare dei valori dei parametri che rendano gli scarti, nel loro insieme, i più piccoli possibili.

Dei due coefficienti, quello che ha veramente importanza è a, cioè il coefficiente direttivo della retta, poiché b dipende dalla scelta dell'origine degli assi e non ha un significato statistico. Se, ad esempio, a fosse uguale a 0, ciò significherebbe che Y è indipendente da X. È interessante notare che spesso X e Y sono grandezze diverse e quindi a non è un numero fisso, ma una grandezza il cui valore numerico dipende dalle unità di misura prescelte. Il coefficiente a prende il nome di coefficiente di regressione lineare di Y rispetto ad X. Il coefficiente di regressione lineare è dato dal rapporto fra la covarianza e la varianza della variabile indipendente.

Se si vuole studiare la dipendenza di X da Y, si procede in modo analogo al precedente, ponendo $x=a'y+b'$ e trovando i due parametri sempre con il metodo utilizzato per il coefficiente di regressione lineare. Se la dipendenza tra le due variabili fosse perfetta, risulterebbe evidentemente $a'=1/a$ e la retta coinciderebbe con la precedente; in generale, però, ciò non avviene, perché nel primo caso si minimizza la sommatoria degli scarti $(y'-y)^2$, nel secondo caso la sommatoria $(x'-x)^2$, cioè le somme dei quadrati delle distanze dei punti osservati dalla retta interpolatrice, misurate nel senso delle y e delle x. L'espressione di a' diviene a' = rapporto fra la covarianza e la varianza della variabile indipendente (che in questo caso è la y). Il numeratore non varia, perché evidentemente la espressione della covarianza non muta scambiando X con Y.

Coefficiente di correlazione lineare

L'indice quadratico di dipendenza visto, permette di misurare il grado di adattamento di una variabile statistica doppia ad una curva di dipendenza di forma e tipo qualsiasi, purché ben determinata e definita anche numericamente. La curva di dipendenza deve cioè essere una funzione calcolabile numericamente per ogni valore della x.

Analizziamo ora in quale grado la dipendenza fra gli argomenti X e Y di una variabile statistica doppia sia ben rappresentata da una curva di cui si stabilisce solo la forma senza fissarne i parametri. Particolare interesse riveste il caso in cui la famiglia delle curve di dipendenza sia costituita da rette e perciò il legame fra X e Y sia di tipo lineare.

Si introduce un indice chiamato coefficiente di correlazione lineare, che misura il grado di adattamento dei dati sperimentali all'ipotesi che fra gli argomenti X e Y della variabile statistica esista un legame di tipo lineare.

La correlazione è un aspetto più generale della regressione, in quanto considera la relazione intercedente tra le due variabili, senza porre la condizione di dipendenza di natura causale tra le

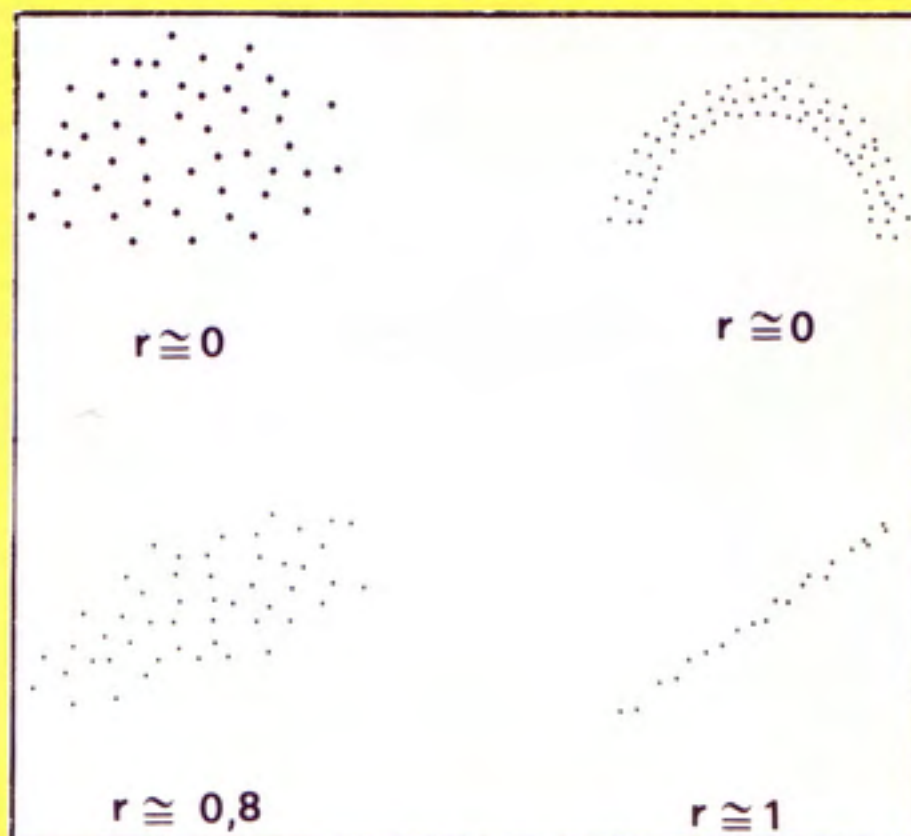



Figura 4 - Valori del coefficiente di correlazione r per vari insiemi di punti.

variabili stesse. In altre parole, nella correlazione viene considerata l'interdipendenza tra le variabili, in quanto sia una che l'altra possono fungere da variabile indipendente.

Il coefficiente di correlazione lineare ha un valore sempre compreso tra −1 e +1: uguale a ±1 solo quando si ha perfetta correlazione lineare, uguale a zero nei casi di perfetta indipendenza di tipo lineare fra X e Y, e compreso fra +1 e 0 oppure fra 0 e −1 in tutti gli altri casi. Il coefficiente r è dato dal rapporto tra la differenza $M_{xy}-M_xM_y$ ed il prodotto della varianza di x per la varianza di y. Il calcolo del valore M_{xy} viene eseguito mediante la seguente espressione:

$$M_{xy} = \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{m} X_i Y_i f_{ij}$$

A differenza dell'indice di Pearson, che è una misura generale della dipendenza esistente tra i due argomenti della variabile, il rapporto r ha significato solamente quando la dipendenza ha un andamento pressoché rettilineo. Esso, infatti, è un indice che mette in luce la maggiore o minore approssimazione della retta come rappresentatrice della correlazione fra gli argomenti della variabile statistica stessa. Esso è inoltre esprimibile come la media geometrica di a e a', i quali sono così uno l'inverso dell'altro solo quando si ha perfetta dipendenza lineare (r=1).

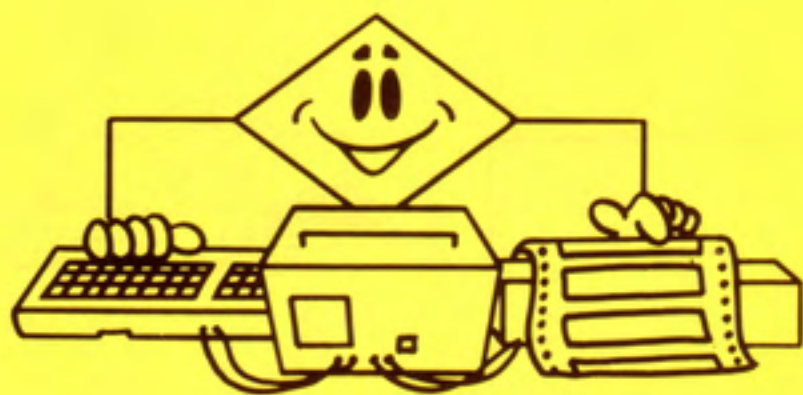
In figura 4 sono rappresentati sul piano X, Y dei casi di variabili statistiche a due dimensioni; i punti rappresentano ciascuno un individuo della popolazione in esame. Per ognuno dei quattro casi illustrati è dato il valore del coefficiente di correlazione r, che va da 0 nel primo caso ad 1 per l'ultimo.

È interessante notare come, nel secondo caso, pur essendo evidente la correlazione esistente fra X e Y, r sia nullo, poiché il legame fra i due argomenti non è affatto rappresentabile con una retta.

Conclusione

Ricordo, ancora una volta, che mi sono servito del mio testo universitario di statistica, "Fondamenti di Statistica", della prof. Giovanna Togliatti, edito dalla Hoepli, al quale invito a fare riferimento chiunque fosse interessato ad approfondire gli argomenti trattati.

Ricordo, inoltre, che sul prossimo numero di Bit compariranno il listato del programma ed i commenti al funzionamento dello stesso, completi di tutte le istruzioni necessarie ad un corretto utilizzo delle opzioni che esso offre.

(Continua)

ZX SPECTRUM

Prima e Terza: due programmi didattici per Spectrum

Abbiamo visto e letto di Scuole elementari ove maestri convinti, entusiasti, addirittura pervicaci, stanno lavorando con l'elaboratore. Abbiamo anche preso visione di diagrammi di flusso, di programmi che gli scolari sono arrivati in breve tempo a stendere. Se da un lato ciò dimostra che l'entusiasmo del maestro è molla fondamentale nel processo di apprendimento dello scolaro, da un altro canto ci ha portati a meditare se non sia il caso di accostarci all'informatica usando la prudenza dei piccoli passi. Certamente gli elaboratori possono trovare ampio spazio nel campo educativo, ma in più di un filone di intervento.

di **Dino Citterio**

Prima

Non crediamo si possa mettere in dubbio che il computer favorisce l'apprendimento, anzi ci sentiamo di aggiungere che proprio le procedure di lavoro che impone sono il suo pregio maggiore. Ci impone di aver ben chiaro l'obiettivo da raggiungere, pretende l'analisi dei dati, avvia alla soluzione attraverso l'algoritmo. Se riflettiamo un attimo, cambiati i termini, è quanto chiediamo agli scolari, sottoponendo loro il classico problema da risolvere. Lo scolaro deve esporre i dati, ideare una strategia di elaborazione degli stessi e spiegare il motivo delle varie operazioni, quindi riassumere i risultati ottenuti sotto forma di una risposta ad un quesito iniziale. Ciò significa che, almeno con l'aritmetica, siamo vicini all'informatica più di quanto non si creda. Non vogliamo ora allargare ulteriormente il discorso, passiamo a giustificare il programma. È nato per i bambini che in prima elementare (e magari anche più in là) ancora incontrano difficoltà di calcolo nell'ambito del 20. L'abbiamo sviluppato in due aspetti: con il primo la macchina si propone all'alunno come fonte di esercitazione; presenta l'operazione, controlla il risultato introdotto ed evidenzia, in caso di errore, l'operazione proposta usando la grafica.

Apportate al programma le modifiche che verranno consigliate, potremo usare lo stesso in funzione di controllore. Dati e risultati sono introdotti con INPUT dall'alunno e la macchina si sostituisce ancora di più all'insegnante nella fase della correzione. Perché il computer invece del maestro? Soprattutto perchè l'alunno non rimane passivo ad aspettare la spiegazione, ma deve correggere il proprio errore osservando lo schermo. Anche in passato gli alunni controllavano il loro lavoro con materiale strutturato; i nostri padri hanno usato il pallottoliere (ma è proprio superato?), noi sassi, fagioli e granoturco, i nostri figli i numeri in colore. Alle ultime leve offriamo l'elaboratore, visto che già lo conoscono come fonte di gioco. Esso ha eliminato la manipolazione del materiale e quindi tutti gli errori legati ad essa, chiede che si risponda premendo tasti a ragion veduta ed infine (è importante) corregge, proponendo sempre lo stesso schema grafico. Lo stimolo che ogni volta viene proposto dallo schermo dovrebbe contribuire a fissare la procedura di lavoro, il metodo da seguire.
Il programma in prima versione può essere detto di rinforzo. Propone l'operazione (linee 115, 215, ...), verifica l'esattezza del risultato (linea 120, ...), visualizza graficamente in caso di errore (linea 126 e seguito).

Non ci sembra il caso di diffonderci in spiegazioni perché il programma in sé è piuttosto semplice. Facciamo solo rilevare che le operazioni vengono proposte a gruppi di dieci (contate con la variabile "e"), le risposte esatte premiate con un palloncino appeso nella parte alta dello schermo, dopo di che (linea 610) si può ritornare alla linea 72 ed effettuare una scelta di lavoro diversa. Si può alleggerire il listato eliminando lo schermo iniziale di presentazione fino alla linea 49 esclusa.
Vediamo ora le varianti. Con le sostituzioni che seguono il programma assume veste tutoria. L'alunno, dopo aver effettuato le sue operazioni, le controlla con l'aiuto di Spectrum come già detto. Il vantaggio di poter contare su più di una visualizzazione consente di andare incontro meglio alle difficoltà del bambino e permette anche di adattare il lavoro del computer ad eventuali accorgimenti seguiti dall'insegnante in fase di impostazione dell'argomento. Si pensi all'aritmetica insegnata con i numeri in colore, oppure con la pista dei numeri, col materiale Meina, e così via.

Modifiche

Uso del programma come controllo di addizioni entro il 20 e con due addendi. Sostituire le linee di pari numero con le seguenti:
105 INPUT "Dammi il primo addendo"; a
107 INPUT "Dammi l'altro addendo"; b
La linea 110 può essere tolta, perché si dà per scontato che il totale non supererà il 20.
Uso del programma per controllare le sottrazioni entro il 20 con minuendo e un solo sottraendo:
205 INPUT "Dammi il minuendo"; a

ZX SPECTRUM

207 INPUT "Dammi il sottraendo"; b
La linea 210 può essere tolta perché si suppone dato un minuendo maggiore del sottraendo.
Uso del programma per controllare le moltiplicazioni.
305 INPUT "Dammi il moltiplicando"; a
307 INPUT "Dammi il moltiplicatore"; b
Per il controllo delle divisioni si consiglia di scegliere il caso "di contenenza".
760 INPUT "Dammi il dividendo"; a
762 INPUT "Dammi il divisore"; b
È possibile togliere la linea 764.
Può rivelarsi utile anche questa variante, che permette di visualizzare entrambi gli addendi sulla stessa linea di schermo:
138 PRINT INK 2; AT 13,x;"*": BEEP .5,12: NEXT x.
Si noti l'introduzione del BEEP; si è voluto rallentare di proposito l'operazione di "aggiungere", in modo che il bambino vedesse chiaramente l'addendo aggiunto ad un'unità per volta.
Per la sottrazione consigliamo questa variante, che allinea a destra il sottraendo, così come si fa quando si mette in colonna. Il BEEP ha la stessa funzione già illustrata più sopra.
237 FOR x=6+a TO x-b STEP −1
238 PRINT OVER 1; INK 2; AT 13,x;"*":BEEP; .5,12
240 FOR y = 7 TO x
245 mettere y invece di x
248 NEXT y

Terza

È del tutto logico, dopo aver caricato sullo Spectrum un programma di aritmetica per la figlia che frequenta la prima elementare (calcolo entro il 20), sentirsi chiedere da quella di terza elementare: "E a me non lo fai il programma?. Come no! Vedilo in figura 2".
Per i lettori diremo che non è il caso di spendere molte osservazioni per illustrare la programmazione. Per la didattica diciamo che si è scelta la strada più logica, praticamente l'unica percorribile quando si avviano gli scolari alla tecnica del calcolo mentale. L'insegnante chiede sempre la scomposizione del numero in decine ed unità, la loro somma, l'eventuale riporto della decina quando la somma delle unità supera il 9. Nella sottrazione, dopo la scomposizione, può essere necessario lo spostamento della decina fra le unità. Per meglio rendere visibile il procedimento si è arrestato più volte il programma con PAUSA 0 invitando l'alunno a premere un tasto per proseguire solo dopo aver capito la fase illustrata sullo schermo. In caso di ulteriore errore la dimostrazione viene interamente ripetuta. Il discorso si applica anche all'addizione quando la somma delle unità supera il 9. Anche in questo programma si sono distinti i due casi di divisione: contenenza e ripartizione. È opportuno che i due concetti siano ben chiari ai bambini (qualche collega ha un programma da offrire in proposito?), diversamente è bene scegliere il caso di contenenza perché ad esso si fa riferimento quando si esegue il calcolo. Noi siamo soliti dire "quante volte il 5 sta nel 26" e solo all'atto del problema ci troviamo nella necessità di dividere una quantità in parti uguali, cercando la consistenza di ciascuna parte.

Le operazioni vengono proposte a gruppi di 20, con eventuale ritorno alla scelta di lavoro (linea 700). Anche questo programma può facilmente essere modificato per utilizzarlo come controllo di operazioni di cui siano noti i due dati. Basta intervenire sulle linee che assegnano i valori ad "a" e "b" e sostituirle con INPUT così come spiegato nel programma "prima".

```
   1 OVER 1
   5 PAPER 5: INK 1: BORDER 1: C
LS
  12 LET x=128
  15 LET y=87
  18 FOR j=0 TO 255
  20 PLOT x,y: DRAW j-x,-y
  23 PLOT x,y: DRAW j-x,175-y
  25 BEEP .1,INT (j/5)
  26 NEXT j
  28 FOR k=0 TO 175
  30 PLOT x,y: DRAW -x,k-y
  32 PLOT x,y: DRAW 255-x,k-y
  35 BEEP .03,k/3: BEEP .01,j/5:
 NEXT k
  37 PAUSE 100
  39 BORDER 6: OVER 0
  40 PRINT PAPER 7; BRIGHT 1;AT
10,5;"DIDATTICA E CALCOLATORE"
  45 PAUSE 150
  50 PAPER 7: BORDER 3: INK 0: C
LS
  53 PRINT TAB 5;"ESERCIZI DI AR
ITMETICA"
  55 FOR x=19 TO 4 STEP -1
  68 PRINT AT x,6;"CALCOLO ENTRO
 IL 20"
  70 NEXT x
  71 PAUSE 150
  72 CLS : PRINT INK 3;"[illegible]
[illegible]";AT 6,0
;"[illegible]"
  73 LET e=0: LET s=0
  74 PRINT AT 3,8;"Scelta di lav
oro"
  75 PRINT AT 16,0;"(Ti faro' fa
re 10 calcoli.      Ad ogni risu
ltato esatto appen- dero' un pal
loncino al filo che vedrai sotto
 il titolo.)"; BRIGHT 1;AT 21,0;
"Premi un tasto per incominciare
."
  76 PRINT AT 8,0;"Per fare le A
DDIZIONI   (premi A)"
  78 PRINT AT 10,0;"Per fare le
SOTTRAZIONI(premi S)"
  79 PRINT AT 12,0;"Per le MOLTI
PLICAZIONI (premi M)"
  80 PRINT AT 14,0;"Per fare le
DIVISIONI   (premi D)"
  81 PAUSE 0
  82 IF INKEY$="s" THEN GO TO 20
0
  85 IF INKEY$="m" THEN GO TO 30
0
  87 IF INKEY$="d" THEN GO TO 40
0
 100 CLS : LET g=101
 101 PRINT AT 0,11;"ADDIZIONI"
 102 GO SUB 750
 105 LET a=INT (20*RND)+1
 107 LET b=INT (20*RND)
 110 IF a+b>20 THEN LET b=ABS (b
-INT (1+3*RND)): GO TO 110
 115 PRINT AT 4,0;"Dimmi quanto
fa ";a;"+";b
 118 INPUT "Risultato= ";r
 120 IF r=a+b THEN PRINT PAPER 6
;AT 4,22;r: GO TO 500
 122 GO SUB 520
 126 IF a>=10 THEN PRINT AT 13,3
;a
 127 IF a<10 THEN PRINT INK 1;AT
 13,4;a
 130 FOR x=7 TO 6+a
 132 PRINT INK 1;AT 13,x;"#": NE
XT x
 133 IF b>=10 THEN PRINT AT 14,2
;"+";b
 135 IF b<10 THEN PRINT AT 14,2;
"+ ";b
 137 FOR x=7+a TO 6+a+b
 138 PRINT INK 2;AT 14,x;"*": NE
XT x
 150 LET d=INT (a+b)/10
 155 LET u=a+b-INT d*10
 160 GO SUB 700
 175 GO TO 118
 200 CLS : BORDER 5: LET g=201
 201 PRINT AT 0,11;"SOTTRAZIONI"
 202 GO SUB 750
 205 LET a=INT (6+20*RND): IF a>
20 THEN LET a=20
 207 LET b=INT (20*RND)
 210 IF a<b THEN LET b=b-INT (1+
```

Listato 1 - Listato programma "Prima".

ZX SPECTRUM

Seguito programma "Prima".

```
3*RND): GO TO 210
 215 PRINT AT 4,0;"Dimmi quanto
fa ";a;"-";b
 218 INPUT "Risultato= ";r
 220 IF r=a-b THEN PRINT PAPER 6
;AT 4,22;r: GO TO 500
 222 GO SUB 520
 226 IF a>=10 THEN PRINT INK 1;A
T 13,3;a
 227 IF a<10 THEN PRINT INK 1;AT
 13,4;a
 230 FOR x=7 TO 6+a
 232 PRINT INK 1;AT 13,x;"#": NE
XT x
 233 IF b>=10 THEN PRINT INK 2;A
T 14,2;"-";b
 235 IF b<10 THEN PRINT INK R;AT
 14,3;"-";b
 237 FOR x=7 TO 6+b
 238 PRINT INK 2;AT 14,x;"*"
 239 NEXT x
 240 FOR x=7+b TO 6+a
 245 PRINT FLASH 1;AT 16,x;"+"
 248 NEXT x
 250 LET d=INT (a-b)/10
 255 LET u=a-b-INT d*10
 258 GO SUB 700
 260 GO TO 218
 300 CLS : BORDER 4: LET g=301
 301 PRINT AT 0,8;"MOLTIPLICAZIO
NI"
 302 GO SUB 750
 305 LET a=INT (6*RND)
 307 LET b=INT (5*RND)
 315 PRINT AT 4,0;"Dimmi quanto
fa ";a;"x";b
 318 INPUT "Risultato= ";r
 320 IF r=a*b THEN PRINT PAPER 6
;AT 4,22;r: GO TO 500
 322 GO SUB 520
 323 FOR w=11 TO 16: PRINT AT w,
3;"                          ": N
EXT w
 324 PRINT AT 10,0;a;"x";b;TAB 1
1;"oppure  ";b;"x";a
 328 FOR x=4 TO 3+a
 330 FOR y=10 TO 9+b
 332 PRINT INK 2;AT y,x;"#"
 335 NEXT y
 340 NEXT x
 345 FOR x=26 TO 25+b
 348 FOR y=10 TO 9+a
 350 PRINT INK 1;AT y,x;"*"
 352 NEXT y
 355 NEXT x
 360 LET d=INT (a*b)/10
 363 LET u=a*b-INT d*10
 370 GO SUB 700
 375 GO TO 318
 400 BORDER 6: CLS : GO SUB 750
 401 PRINT AT 4,8;"SCELTA DI LAV
ORO"
 402 PRINT AT 11,0;"DIVISIONI di
 CONTENENZA -premi C";AT 13,0;".
...... di RIPARTIZIONE -premi R"
: BRIGHT 1;AT 21,0;"Premi un tas
to per incominciare.": PAUSE 0
 403 IF INKEY$="r" THEN GO TO 45
0
 404 CLS : BORDER 3: LET g=405
 405 PRINT AT 0,4;"DIVISIONI DI
CONTENENZA"
 406 GO SUB 750
 410 GO SUB 760
 415 INPUT "Risultato ";r,"Resto
 ";s
 420 IF r=d AND s=o THEN PRINT P
APER 6;AT 4,22;d;" Resto ";o: GO
 TO 500
 423 GO SUB 800
 425 PRINT AT 12,0;a;":";b
 428 FOR x=80 TO 80+(8*b*d) STEP
 (8*b)
 429 PLOT x,66: DRAW 0,5
 430 NEXT x
 435 PRINT AT 18,0;"Quante volte
 ";a;AT 19,3;"contiene ";b; "?"
 438 NEXT x
 446 GO TO 415
 450 REM divisione di ripartizio
ne
 451 CLS : LET g=452
 452 PRINT AT 0,3;"DIVISIONE DI
RIPARTIZIONE"
 453 GO SUB 750
 455 GO SUB 760
 458 INPUT "Risultato ";r,"Resto
 ";s
 464 IF r=d AND s=o THEN PRINT P
APER 6;AT 4,22;d;" Resto=";o: GO
 TO 500
 466 GO SUB 800
 470 PRINT AT 12,2;a;AT 13,1;":"
 b;" parti"
 472 FOR x=80 TO 80+(8*b*d) STEP
 8*d
 475 PLOT x,66: DRAW 0,5
 478 NEXT x
 480 PRINT AT 18,5;"Quanti #";AT
 19,1;"in ogni parte?"
 485 NEXT x
 490 GO TO 458
 500 PRINT AT 8,0;"************
ESATTO ************": BEEP 1,11:
 BEEP 2,15: PRINT AT 8,0;"
                              "
 501 CLS
 502 LET e=e+2: IF e=20 THEN GO
TO 600
 503 FOR x=6 TO 5+e STEP 2: PRIN
T AT 2,x;"©": NEXT x
 505 GO TO g
 520 PRINT FLASH 1;AT 8,0;"▪▪▪▪▪
▪▪▪▪▪▪▪ ERRORE ▪▪▪▪▪▪▪▪▪▪▪▪▪": BE
EP 3,-15: PRINT BRIGHT 1;AT 8,0;
"OSSERVA E CALCOLA CON ATTENZION
E"
 522 LET s=s+1
 525 PRINT PAPER 6;AT 11,7;"  DE
CINA  "; PAPER 5;"  DECINA   "
 528 PRINT AT 17,0;"
"
 529 PRINT AT 18,9;"Quante     Qu
ante";AT 19,8;"DECINE?   UNITA'?
";AT 21,0;"totale"
 530 PLOT 55,0: DRAW 0,87: PLOT
135,0: DRAW 0,87: PLOT 217,0: DR
AW 0,87
 535 RETURN
 600 CLS : PRINT "Hai eseguito 1
0 calcoli.",,,"Gli errori sono s
tati ";s
 605 PRINT AT 10,0;"Se il tuo la
voro e' finito premi il tasto F"
'''"Se vuoi continuare i calcoli
    premi C": PAUSE 0
 610 IF INKEY$="c" THEN GO TO 72
 620 CLS : PRINT AT 10,5;"Arrive
derci a presto."
 625 STOP
 700 FOR x=10 TO 9+d
 705 PRINT BRIGHT 1;AT 21,x;"|"
 710 NEXT x
 713 FOR x=17 TO 16+u
 715 PRINT INK 3;AT 21,x;"*"
 718 NEXT x
 720 RETURN
 750 PRINT AT 1,0;"}}}}}}| | | |
 | | | | | | {{{{{{";AT 6,0;"}}}
}}}}}}}}}}}}}}{{{{{{{{{{{{{{{{"
 755 PLOT 48,167: DRAW 160,0: RE
TURN
 760 LET a=INT (4+19*RND): IF a>
20 THEN LET a=20
 762 LET b=INT (10*RND)+1
 764 IF a<b THEN LET b=b-INT (2+
2*RND): GO TO 764
 766 LET d=INT (a/b): LET o=a-b*
d
 767 PRINT AT 4,0;"Dimmi quanto
fa ";a;":";b
 770 RETURN
 800 IF r<>d AND s=o THEN PRINT
AT 8,0;"  Hai sbagliato il risu
ltato.  "
 801 IF r=d AND s<>o THEN PRINT
AT 8,0;"     Hai sbagliato il re
sto     "
 802 IF r<>d AND s<>o THEN PRINT
 AT 8,0;"  Risultato e resto sb
agliati  "
 803 BEEP 5,-24: PRINT BRIGHT 1;
AT 8,0;"OSSERVA E CONTROLLA ATTE
NTAMENTE"
 804 PRINT PAPER 6;AT 11,10;"  d
ecina  "; PAPER 5;"  decina   "
 805 PRINT AT 17,0;"
"
 807 PLOT 128,0: DRAW 0,36
 810 FOR x=10 TO 9+a
 815 PRINT INK 1;AT 12,x;"#": NE
XT x
 820 PLOT 80,66: DRAW 8*b*d,0
 823 FOR x=10+b*d TO (9+a)
 825 PRINT FLASH 1;AT 14,x;"+"
 830 NEXT x
 835 PRINT AT 18,17;"Con resto d
i?"
 838 FOR x=0 TO (d-1): PRINT BRI
GHT 1; INK 1;AT 21,x;"|"
 840 NEXT x
 844 FOR x=17 TO (16+o): PRINT I
NK 2;AT 21,x;"+"
 846 NEXT x
 850 RETURN
9000 CLS : PRINT FLASH 1;AT 10,0
;"         FERMA IL NASTRO
  "
9005 PRINT AT 21,0;"Premi un tas
to per incominciare.": PAUSE 0
9010 RUN
```

Listato 2 - Listato programma "Terza".

```
  5 PLOT 144,136: DRAW -32,0,3.
14*PI
  7 PLOT 128,46: DRAW 0,72
 10 PLOT 88,0: DRAW 24,0: DRAW
16,46: DRAW 16,-48: DRAW 24,0
 15 PLOT 64,160: DRAW 64,-80: D
RAW 64,80
 20 PLOT 120,144: PLOT 136,144
 25 PLOT 124,132: DRAW 8,0,PI/2
 30 PLOT 125,136: DRAW 6,0
 35 PLOT 128,136: DRAW 0,8
 40 PRINT BRIGHT 1;AT 1,4;"DIDA
TTICA E CALCOLATORE"
 49 PAUSE 100
 50 CLS : PRINT TAB 10;"Classe
TERZA"
 51 PRINT AT 3,0; PAPER 6;"   E
SERCITAZIONE DI ARITMETICA  "
 52 PRINT AT 8,7;"Scelta di lav
oro"
 55 GO SUB [illegible]
 58 PRINT AT 11,0;"Per fare le
ADDIZIONI    premi A"'"Per fare
le SOTTRAZIONI  premi S"'"Per le
 MOLTIPLICAZIONI   premi M"'"Per
 fare le DIVISIONI     premi D"
 59 PRINT AT 17,0;" Ti chiedero
' 20 operazioni e adogni rispost
a esatta appendero' un palloncin
o sotto  il titolo. ": GO SUB 60
0
 60 PAUSE 0
 62 IF INKEY$="s" THEN GO TO 20
0
 64 IF INKEY$="m" THEN GO TO 30
0
 65 IF INKEY$="d" THEN GO TO 40
0
100 CLS : BORDER 5: LET e=0: LE
T z=101
101 PRINT AT 0,11;"ADDIZIONI"
102 GO SUB 500
105 LET a=INT (99*RND): LET b=I
NT (99*RND)
107 IF a+b>=100 THEN LET b=b-IN
T (10*RND): GO TO 107
108 IF b<0 THEN LET b=0
110 PRINT AT 3,0;"Dimmi quanto
fa ";a;"+";b
112 INPUT "Risultato ";r
115 IF r=a+b THEN GO TO 510
120 GO SUB 523
121 GO SUB 525
123 PRINT AT 14,0;"+"
125 FOR x=4 TO 3+h+k
127 PRINT AT 17,x;"#": NEXT x
130 FOR x=14 TO 13+s+t
133 PRINT AT 17,x;"*": NEXT x
135 IF s+t>9 THEN PRINT AT 17,4
+h+k;" ": PRINT FLASH 1;AT 19,14
"Attenzione Riporto": GO SUB 60
0: PAUSE 0: FOR x=14 TO 23: PRIN
T BRIGHT 1;AT 17,x;"*": NEXT x:
PAUSE 0: PRINT BRIGHT 1;AT 17,4+
h+k;"#": PRINT AT 17,14;"
          "
145 GO SUB 620
148 GO TO 112
200 CLS : BORDER 6: LET e=0: LE
T z=201
201 PRINT AT 0,11;"SOTTRAZIONI"
205 GO SUB 500
206 LET a=INT (10+90*RND): LET
b=INT (2+89*RND)
207 IF a<b THEN LET b=b-10: GO
TO 207
210 PRINT AT 3,0;"Dimmi quanto
fa ";a;"-";b
212 INPUT "Risultato ";r
215 IF r=a-b THEN GO TO 510
220 GO SUB 523
222 GO SUB 525
223 PRINT AT 14,0;"-"
224 PRINT AT 12,14+s;"
 ";AT 17,4;"             ";AT 17,14;
'                 "
225 IF s>=t THEN GO TO 228
227 PRINT PAPER 5;AT 19,0;"Atte
nzione! ";s;"-";t;" non si puo'f
are.Debbo dare 1 decina alle uni
ta'.": GO SUB 600: PAUSE 0: PRIN
T BRIGHT 1;AT 12,3+h;"#": PAUSE
0: PRINT AT 12,3+h;" ": PRINT BR
IGHT 1;AT 12,14+s;"**********":
PAUSE 0: PRINT AT 12,3+h;" ": LE
T h=h-1: LET s=s+10
228 FOR x=14+t TO 13+s: PRINT A
T 17,x;"*": NEXT x
230 FOR x=4+k TO 3+h
233 PRINT AT 17,x;"#": NEXT x
235 GO SUB 620
240 GO TO 212
300 CLS : BORDER 5: LET e=0: LE
T z=301
301 PRINT AT 0,8;"MOLTIPLICAZIO
NI"
302 GO SUB 500
305 LET a=INT (11*RND): LET b=I
NT (11*RND)
307 PRINT AT 3,0;"Dimmi quanto
fa ";a;"x";b
310 INPUT "Risultato ";r
313 IF r=a*b THEN GO TO 510
315 GO SUB 523
318 PRINT AT 9,3;a;"x";b;"
oppure";AT 9,25;b;"x";a
320 FOR x=11 TO 10+b
323 FOR y=1 TO 0+a
325 PRINT INK 1;AT x,y;"#"
328 NEXT y
330 NEXT x
333 FOR x=11 TO 10+a
335 FOR y=21 TO 20+b
337 PRINT INK 2;AT x,y;"#"
338 NEXT y
340 NEXT x
342 GO SUB 620
345 GO TO 310
400 CLS : PRINT : PRINT ;TAB 4;
"Hai scelto le divisioni"
401 PRINT AT 8,5;"(C) ... di co
ntenenza?": PRINT AT 10,5;"(R)
 di ripartizione?"
402 PRINT BRIGHT 1;AT 20,0;" Pr
emi il tasto corrispondente. "
403 PAUSE 0
404 IF INKEY$="c" THEN GO TO 80
0
405 CLS : BORDER 5: LET e=0: LE
T z=406
406 PRINT AT 0,4;"DIVISIONE DI
RIPARTIZIONE"
408 GO SUB 500
410 LET a=INT (12+88*RND): LET
b=INT (a/10+(11-a/10)*RND)
411 LET h=INT (a/b)
413 LET k=a-h*b
415 PRINT AT 4,5;"Dimmi quanto
fa ";a;":";b
416 INPUT "Risultato ";r,"Resto
";o
418 IF r=h AND k=o THEN GO TO 5
05
420 GO SUB 520
422 PRINT AT 8,0;a;":";b;"="
423 FOR y=8 TO 7+2*b STEP 2
425 FOR x=8 TO 7+h
430 PRINT AT x,y; PAPER 6;"#"
433 NEXT x
435 NEXT y
445 FOR y=8 TO 7+2*k STEP 2
450 PRINT INK 2;AT 9+h,y;"#"
453 NEXT y
460 PRINT AT 21,0; PAPER 6;"Qua
nti # nelle colonne colorate?"
465 GO TO 416
500 PRINT INK 1;AT 1,0;"[illegible]
||||||||||||||||||||[illegible]";AT 5,
0;"[illegible]
[illegible]": PLOT 48,167: DRAW 160,0: R
ETURN
505 PRINT BRIGHT 1;AT 21,0;"
                              ";A
T 21,0;"Esatto. Risulta ";r;". C
on resto ";k
510 PRINT BRIGHT 1;AT 21,0;"ESA
TTO. Risulta ";r: BEEP .5,0: BEE
P 1,5: BEEP .5,9: BEEP .5,5: BEE
P 1.5,12
512 CLS : LET e=e+1: FOR x=6 TO
5+e: PRINT AT 2,x;"©": NEXT x
513 IF e=20 THEN GO TO 700
515 GO TO z
520 IF r<>h AND o=k THEN PRINT
AT 6,0;"   Hai sbagliato il risu
ltato   "
521 IF r=h AND o<>k THEN PRINT
AT 6,0;"     Hai sbagliato il re
sto     "
522 IF r<>h AND o<>k THEN PRINT
 AT 6,0;" Risultato e resto sba
gliati  "
523 PRINT FLASH 1;AT 21,0;"
           ERRORE               "
524 BEEP 5,-24: PRINT BRIGHT 1;
AT 6,0;"OSSERVA E CONTROLLA ATTE
NTAMENTE": RETURN
525 PRINT AT 9,9;"Scompongo"; P
APER 4;AT 10,4;"  decine  "; PAP
ER 6;AT 10,14;"    unita'
"
527 PLOT 31,30: DRAW 0,65: PLOT
 110,30: DRAW 0,65: PLOT 31,68:
DRAW 223,0: PLOT 0,52: DRAW 255,
0: PLOT 0,50: DRAW 255,0: PLOT 0
,29: DRAW 255,0
530 IF a>9 THEN PRINT AT 12,1;a
532 IF a<10 THEN PRINT AT 12,2;
```

ZX SPECTRUM

Seguito programma "Terza".

```
 534 IF b>9 THEN PRINT AT 14,1;b
 535 IF b<10 THEN PRINT AT 14,2;
b
 536 LET h=INT (a/10): LET k=INT
 (b/10)
 537 FOR x=4 TO 3+h
 539 PRINT AT 12,x;"#": NEXT x
 540 FOR x=4 TO 3+k
 542 PRINT AT 14,x;"#": NEXT x
 545 LET s=a-h*10: LET t=b-k*10
 550 FOR x=14 TO 13+s
 552 PRINT AT 12,x;"*": NEXT x
 560 FOR x=14 TO 13+t
 562 PRINT AT 14,x;"*": NEXT x
 565 RETURN
 600 PRINT BRIGHT 1;AT 21,0 ;"
Se hai capito premi un tasto: ":
 RETURN
 520 PRINT AT 21,0;"
                             ": RETURN
 700 BORDER 3: CLS : PRINT TAB 1
0;"Complimenti!"'''"Hai eseguito
20 calcoli esatti.";AT 10,0;"Pre
mi Z per fare altri calcoli"''"S
e hai finito il lavoro premi M"
 705 PAUSE 0
 710 IF INKEY$="z" THEN GO TO 50
 720 PRINT AT 21,2;"Ciao e arriv
ederci a presto."
 725 STOP
 800 CLS : BORDER 4: LET è=0: LE
T z=805
 805 PRINT AT 0,5;"DIVISIONE DI
CONTENENZA"
 810 GO SUB 500
 815 LET a=INT (12+88*RND): LET
b=INT (a/10+(11-a/10)*RND)
 820 LET h=INT (a/b): LET k=a-h*
b
 825 PRINT AT 4,5;"Dimmi quanto
fa ";a;":";b
 830 INPUT "Risultato ";r,"Resto
 ";o
 835 IF r=h AND k=o THEN GO TO 5
05
 840 GO SUB 520
 845 PRINT AT 8,0;a;":";b;"="
 850 FOR y=8 TO 7+2*h STEP 2
 855 FOR x=8 TO 7+b
 860 PRINT PAPER 6;AT x,y;"#"
 864 NEXT x
 866 NEXT y
 870 FOR x=8 TO 7+k
 875 PRINT INK 2;AT x,8+2*h;"#"
 880 NEXT x
 885 PRINT PAPER 6;AT 21,0;" Qua
nte colonne intere ci sono? "
 890 GO TO 830
9000 CLS : PRINT FLASH 1;AT 10,0
;"           FERMA IL NASTRO
  "
9010 PRINT AT 21,0;"Premi un tas
to per incominciare.": PAUSE 0
9020 RUN
```

Sarà a breve disponibile la cassetta del programma Simulavolo. Inviate il tagliando per prenotazione (cod. SP481A/ lire 25.000).

SUL PROSSIMO NUMERO DI SUPER BIT RISERVATO PERSONAL

TROVERETE:

- MAGICATALOG: COME TENERE ORDINATI I DISCHI DELL'APPLE
- TEXT-EDITOR: EDIZIONE DI TESTI PER CASIO 702P
- SCORE: LA PALLACANESTRO SU VIC 20
- MONITOR PER C 64: PROGRAMMAZIONE IN LINGUAGGIO MACCHINA RESA COMODA
- L'ITALIA IN VIC: STUDIARE LA GEOGRAFIA TRAMITE IL VIC
- SETTE E MEZZO: GIOCO DIDATTICO PER SPECTRUM
- SPIRITELLI PER TI: LA COMPLETA GESTIONE DEGLI SPRITE SU TI 99/4A
- C/C CON ZX81: GESTIONE DEL CONTO CORRENTE TRAMITE SINCLAIR ZX81

VIC 20

Timer per VIC e C 64

Non vi capita mai di sedervi davanti al video, magari dopo pranzo, pensando "giusto per dare una ritoccatina al programma"? Ebbene, durante questi lavori capita spesso di perdere la nozione del tempo e non ci si accorge che le ore scorrono, spesso con risultati disastrosi specie avendo molti impegni. Quante volte sarà capitato? Certamente molte. Ora, finalmente, il programma Timer sarà un fedele guardiano del vostro tempo.

di **Renato Comini**

Listato 1 - Programma Timer per VIC 20.

```
 90 GOSUB195
 100 PRINT"■SCEGLI L'ORA"
 110 PRINT"■ HHMMSS"
 120 INPUT"■  000000■■■■■■■■";TI$
 130 POKE953,PEEK(160)
 140 POKE954,PEEK(161)
 150 PRINT"■■CHE ORE SONO"
 160 PRINT"■ HHMMSS"
 170 INPUT"■  000000■■■■■■■■";TI$
 180 PRINT"■":SYS826:END
 195 FORI=826TO953:READX:POKEI,X:NEXT:R
ETURN
 200 DATA120,173,20,3,141,183,3,173
 210 DATA21,3,141,184,3,169,83,141
 220 DATA20,3,169,3,141,21,3,88
 230 DATA96,173,160,0,205,185,3,208
 240 DATA89,173,161,0,205,186,3,208
 250 DATA81,169,133,141,17,30,169,167
 260 DATA141,18,30,169,143,141,19,30
 270 DATA169,146,141,20,30,169,129,141
 280 DATA21,30,169,15,141,14,144,169
 290 DATA139,141,10,144,166,255,164,255
 300 DATA136,208,253,202,208,248,169,0
 310 DATA141,14,144,120,173,183,3,141
 320 DATA20,3,173,184,3,141,21,3
 330 DATA169,0,141,17,150,141,18,150
 340 DATA141,19,150,141,20,150,141,21
 350 DATA150,88,76,191,234,255,0,255


Lista simboli grafici

   100 : 1  SHIFT HOME =CHR$(147)

   110 : 1  CRSR↑ =CHR$(17)

   120 : 1  CRSR↑ =CHR$(17)
         8  SHIFT CRSR← =CHR$(157)

   150 : 1  CRSR↑ =CHR$(17)
         1  CRSR← =CHR$(29)

   160 : 1  CRSR↑ =CHR$(17)

   170 : 1  CRSR↑ =CHR$(17)
         8  SHIFT CRSR← =CHR$(157)

   180 : 1  SHIFT HOME =CHR$(147)
```

Lanciato il programma va innanzitutto "caricata" la sveglia, dopodiché viene chiesta l'ora del momento. Questo è tutto: potrete ora impazzire con i games o studiare programmi o comporre articoli senza più preoccuparvi minimamente del tempo. Inesorabile e preciso, ed al momento giusto, il vostro calcolatore vi chiama: si udrà un suono ed in contemporanea nell'angolo in alto a destra dello schermo comparirà l'avvertimento "è ora". Saprete così che è giunto il momento, ahimé, di abbandonare il tutto per dedicarvi ad altro.

Il programma utilizza una routine assembler che viene posta nel buffer della cassetta ed è chiamata 60 volte al secondo da una routine di interrupt, la stessa che provvede all'aggiornamento del TI$ e ad altri controlli.

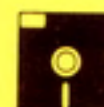
L'algoritmo è molto semplice, si limita ad un confronto tra il predetto orario, in cui deve scattare la sveglia, e l'ora corrente, e dà il segnale al momento giusto. Si noti che la versione per il C 64 differisce da quella del VIC solamente nella parte riguardante le locazioni di memoria relative al buffer della cassetta ed all'interrupt, ma per il resto non vi sono differenze tra i due algoritmi.

Il caricamento del codice assembler, depositato nei DATA che vanno da 200 a 350 per la versione VIC e nei DATA da 200 a 330 per la versione 64, è eseguito dalla linea 195 tramite le istruzioni READ e POKE. Tutta la prima parte del programma (linee 10-180) provvede alla richiesta dell'ora attuale e dell'ora della sveglia.

Attenzione: poiché la routine in linguaggio macchina è depositata nel buffer della cassetta, nessuna operazione con l'uso di questa potrà essere eseguita durante il funzionamento del Timer. Inoltre la pressione del tasto RUN STOP + Restore annulla le chiamate della routine assembler compromettendone il funzionamento, per cui l'allarme non scatta.

VIC 20

```
 10 REM TIMER  C-64
 80 S=54272:FORR=STOS+24:POKER,0:NEXTR
 95 GOSUB195
 100 PRINT"□SCEGLI L'ORA"
 110 PRINT"↓ HHMMSS"
 120 INPUT"↓  000000←←←←←←←←←";TI$
 130 POKE956,PEEK(160)
 140 POKE957,PEEK(161)
 150 PRINT"↓←CHE ORE SONO"
 160 PRINT"↓ HHMMSS"
 170 INPUT"↓  000000←←←←←←←←←";TI$
 180 PRINT"□":SYS49152:END
 195 FORG=49152TO49284:READX:POKEG,X:NE
XTG:RETURN
 200 DATA120,173,20,3,141,186,3,173
 210 DATA21,3,141,187,3,169,25,141
 220 DATA20,3,169,192,141,21,3,88
 230 DATA96,173,160,0,205,188,3,208
 240 DATA92,173,161,0,205,189,3,208
 250 DATA84,169,133,141,17,4,169,167
 260 DATA141,18,4,169,143,141,19,4
 270 DATA169,146,141,20,4,169,129,141
 280 DATA21,4,169,15,141,24,212,169
 290 DATA9,141,5,212,169,6,141,6,212
 300 DATA169,34,141,1,212,169,70,141,0,
212
 305 DATA169,33,141,4,212,169,255,160,2
55,136
 308 DATA208,253,202,208,248,169,0,141,
24,212
 310 DATA120,173,186,3,141,20,3,173,187
,3
 320 DATA141,21,3,88,76,49,234,134,223,
32
 330 DATA223,0,223,32,223,32,223,32,223
,0


Lista simboli grafici


   100 : 1   SHIFT HOME =CHR$(147)

   110 : 1   CRSR↑ =CHR$(17)

   120 : 1   CRSR↑ =CHR$(17)
         8   SHIFT CRSR← =CHR$(157)

   150 : 1   CRSR↑ =CHR$(17)
         1   CRSR← =CHR$(29)

   160 : 1   CRSR↑ =CHR$(17)

   170 : 1   CRSR↑ =CHR$(17)
         8   SHIFT CRSR← =CHR$(157)

   180 : 1   SHIFT HOME =CHR$(147)
```

Listato 2 - Programma Timer per C 64.

ZX 81

di **Sergio Spalletta**

Eatman: il mangione

Come si gioca

Eatman (carattere "E" inverse) va nella buia foresta a caccia di funghi di cui è ghiottissimo. Guidatelo voi con i tasti 5 ed 8 nella sua corsa contro il tempo ed i pericoli cui potrà imbattersi.
Per strada, infatti, troverà funghi velenosi (carattere "X"), da evitare in ogni modo perché sarà subito eliminato, e fantasmi buoni (carattere inverse) che gli regaleranno ognuno un punto.
Quando avrà resistito per 200" (il tempo massimo è di 250"), apparirà nella foresta un folletto matto. Importunatelo perché, se sarà in vena, vi regalerà 10 punti, se sarà arrabbiato, vi eliminerà dal gioco.

Come funziona

Il listato (figura 1) è semplicissimo. Un brevissimo codice macchina (linee 1 - 60) contenuto come di consueto nella 1 REM - formata inizialmente da 7 caratteri qualsiasi - consente di costituire con maggiore rapidità ed economia di memoria la solita istruzione PEEK per la lettura del D-FILE.
Le principali altre routine e subroutine sono:
100 - 121 Inizializzazione delle variabili;
150 - 167 Verifica delle condizioni (valori di U) che determinano la conclusione del gioco;
170 - 195 Routine principale;
200 - 210 Calcolo punteggio fantasmi;
220 Visualizzazione punteggio parziale, tempo e record massimo;
300 - 310 Calcolo punteggio dei funghi (asterischi);
400 - 440 Casualità folletto;
500 - 530 Conclusione;
540 - 710 Presentazione delle opzioni di fine gioco;
2000 - 2100 Presentazione "testata" gioco;
6000 SAVE routine.

Figura 1 - Listato del gioco.

```
  1 REM 0000000
  2 REM ----E A T M A N----
  3 REM BY SERGIO SPALLETTA
  4 REM ----(C)NOV.1983----
  5 POKE 16514,42
 10 POKE 16515,14
 20 POKE 16516,64
 30 POKE 16517,78
 40 POKE 16518,6
 50 POKE 16519,0
 60 POKE 16520,201
 90 GOSUB 2000
100 LET P=9
110 LET M=0
120 LET S=0
121 LET T=0
125 CLS
131 LET T=T+1
132 IF T=250 THEN GOTO 500
133 IF T=20 THEN PRINT AT 21,(R
ND*20)+1;"B"
135 SCROLL

138 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
140 PRINT AT 12,P;
150 LET U=USR 16514
160 IF U<>39 AND U<>8 AND U<>61
 AND U<>151 AND U=139 THEN GOSUB
 200
165 IF U<>39 AND U<>139 AND U<>
151 AND U<>8 AND U=61 THEN GOTO
500
166 IF U<>39 AND U<>8 AND U<>61
 AND U<>139 AND U=151 THEN GOSUB
 300
167 IF U<>8 AND U<>139 AND U<>1
51 AND U<>61 AND U=39 THEN GOSUB
 400
170 PRINT AT 12,P;"E"
180 LET P=P+(INKEY$="8" AND P<2
5)-(INKEY$="5" AND P>0)
190 PRINT AT 21,(RND*25)+1;"■"
191 PRINT AT 21,(RND*25);"X"
192 PRINT AT 21,(RND*25)+1;"*"

195 GOTO 130
197 GOSUB 200
200 LET S=S+1
210 IF S>M THEN LET M=S
220 PRINT AT 0,0;"▒SCORE ";S;"▒
TIME ";T;"▒▒RECORD ";M;"▒▒▒"
230 RETURN
300 LET S=S+5
310 GOSUB 210
320 RETURN
400 LET K=INT (RND*2)+1
410 IF K=1 THEN GOTO 500
420 IF K<>1 THEN LET S=S+10
430 GOSUB 210
440 RETURN
500 PRINT AT 0,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒AAAARGGG▒▒▒"
510 PAUSE 100
520 CLS
530 GOSUB 220
540 PRINT "VUOI RIGIOCARE ?"
550 PRINT "PREMI:"
```

LE VARIABILI PRINCIPALI

K	Controlla l'apparizione (buona o cattiva) del folletto.
P	Posizione di stampa del carattere di Eatman.
M	Record massimo.
S	Punteggio parziale.
T	Tempo trascorso (massimo 250").
U	Posizione sullo schermo di Eatman.

VALORI E PUNTEGGI DEI SIMBOLI

E inverso	Eatman	
★ inverse	Funghi mangerecci	— punti: 5.
" inverse	Fantasmi	— punti: 1.
X	Funghi velenosi	— punti: Eliminazione.
B	Folletto del bosco (valore casuale)	— punti: Eliminazione. — punti: + 10 punti.

Seguito figura 1.

```
 560 PRINT
 570 PRINT "S) PER RIPROVARE
"
 580 PRINT "N) PER SMETTERE  "
 590 PRINT "R) PER RESETTARE IL
RECORD"
 600 PRINT "P) PER SCRIVERE IL T
UO NOME"
 605 PRINT "T) PER REGISTRARE QU
ESTO GIOCO"
 610 IF INKEY$="" THEN GOTO 610
 620 IF INKEY$="S" THEN GOTO 120
 630 IF INKEY$="N" THEN PRINT AT
 15,0;"ARRIVEDERCI ALLA PROSSIMA
VOLTA"
 640 IF INKEY$="N" THEN STOP
 650 IF INKEY$="R" THEN RUN
 660 IF INKEY$="P" THEN PRINT "I
L RECORD ";M;" E'" DI ";
 670 INPUT A$
 680 PRINT A$
 700 GOTO 550

 710 IF INKEY$="T" THEN GOTO 600
0
2000 CLS
2010 PRINT AT 2,0;
2020 PRINT "
  T    E    A
"
2025 PRINT AT 10,0;
2030 PRINT "
  N    M    A
"
2035 PRINT AT 17,0;
2040 PRINT "NELLA NOTTE POPOLATA
 DAI FANTA- SMI IL NOSTRO MANGIO
NE CERCA DI DIVORARNE IL PIU"" P
OSSIBILE,MA  DEVE STARE ATTENTO
ALLE TRAPPOLE"


2050 PAUSE 1000
2100 RETURN
6000 SAVE "EAT-MAN"
6010 RUN
```

```
  E   A   T

  M   A   N

NELLA NOTTE POPOLATA DAI FANTA-
SMI IL NOSTRO MANGIONE CERCA DI
DIVORARNE IL PIU" POSSIBILE,MA
DEVE STARE ATTENTO ALLE TRAPPOLE
```

Figura 2 - Quadro iniziale.

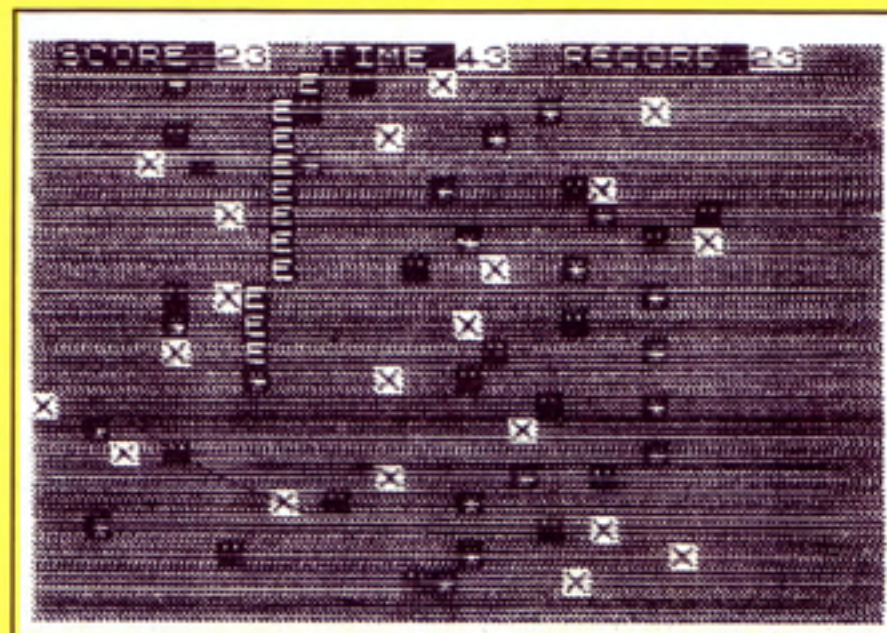
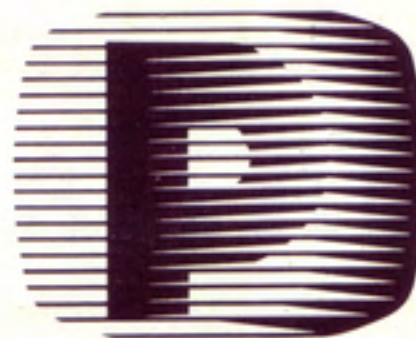



Figura 3 - Copia video di una fase del gioco.

GENERAL PROCESSOR. LA VETTA ITALIANA DEL COMPUTER.

GPS4. LA VETTA

L'hardware.

Costruito con criteri professionali derivanti da una pluriennale presenza sul mercato, l'elaboratore GPS4 può gestire dai floppy 5" da 800 Kbyte ai dischi rigidi da 20 Mbyte, supporta più posti di lavoro, è collegabile alle più diverse periferiche (plotter, digitalizzatori, ecc.) e può trasformarsi lui stesso in terminale intelligente di grossi elaboratori. GPS4 è studiato per crescere insieme alla Vostra Azienda.

GPS4. LA VETTA

Le applicazioni.

Industrie, studi professionali e tecnici, laboratori di analisi, aziende, negozi, alberghi, agenzie pubblicitarie, ed anche scuole, università ed ambienti di ricerca hanno adottato GPS4 per risolvere le loro esigenze di lavoro e di programmazione.
La quota di mercato conquistata da GPS4 è in crescita: in tutti i settori di utilizzo del computer. GPS4 è un successo che cresce.

GPS4. LA VETTA

La bellezza.

Una forma di pura bellezza. Essenziale. Decisa. Una linea di splendido design italiano. GPS4 non è solo perfetto da utilizzare, ma anche bello da guardare. E comodo: la tastiera è separata dallo schermo per consentire la massima fluidità di utilizzo.
GPS4: bello e perfetto.

Vogliate cortesemente inviarmi materiale illustrativo su GPS4.

Nome ....

Cognome ....

Azienda ....

Via ....

Cap .... Città ....

GENERAL PROCESSOR. COMPUTER ITALIANI PER AZIENDE ITALIANE.

GENERAL PROCESSOR

Via del Parlamento Europeo, 9/a - 50010 Badia a Settimo (FI) - Tel.: 055-720.301